University of Alberta

Robust transmission of JPEG2000 images over Rayleigh fading channels

by

Victor Francisco Sanchez Silva

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta

Fall 2002

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-81471-8

Canada

University of Alberta

Library Release Form

Name of Author: Victor Francisco Sanchez Silva

Title of Thesis: Robust transmission of JPEG2000 images over Rayleigh fading channels

Degree: Master of Science

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

10840 81 Av. Apt. 307
Edmonton, Alberta
Canada T6E 1Y4

September 26 , 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Robust transmission of JPEG2000 images over Rayleigh fading channels* submitted by Victor Francisco Sanchez Silva in partial fulfillment of the requirements for the degree of Master of Science.

Dr. Mrinal Mandal

Dr. Xiaodai Dong

Dr. Anup Basu

Sept 26, 2002

Abstract

The JPEG2000 standard has been recently established for coding still images. This standard is likely to be used extensively in the near future. This work addresses the issue of robust transmission of JPEG2000 images over Rayleigh-fading channels. Two unequal channel protection techniques are proposed by taking into account the effect of channel errors of the different layers and packets of the JPEG2000 bit-stream. For a given transmission bandwidth and multi-layered JPEG2000 bit-stream, the proposed techniques adaptively select the best protection scheme according to the channel conditions. The robustness of the proposed techniques is evaluated over a Rayleigh-fading channel with a concatenation of a cyclic redundancy check code and a rate compatible convolutional code. Comparisons are made with the case of no channel protection and equal channel protection when the protection is optimally designed for the channel conditions. Simulation results show a significant improvement on the quality of the received images.

Acknowledgement

I would like to thank my advisor, Dr. Mrinal K. Mandal, for his guidance throughout this research.

This work was supported by the Consejo Nacional de Ciencia y Tecnologia – CONACYT (National Council for Science and Technology), Mexico. I would like to thank CONACYT for the financial support. In addition, thanks to my family, whose support helped bring this thesis to completion.

Table of Contents

1. Introduction **1**

1.1 Error Resilience.....2

1.1.1 Error Resilient Tools.....2

1.1.2 Channel Coding.....3

1.2 Channel Protection for Images.....4

1.3 Motivation.....5

1.4 Major Contributions.....6

1.5 Organization of the Thesis.....8

2. Review of the JPEG2000 Image Coding Standard **10**

2.1 The JPEG2000 Standard Algorithm.....11

2.2 Discrete Wavelet Transform.....13

2.3 Quantization.....18

2.4 Precincts, Layers and Packets.....18

2.5 Bit-stream Formation.....20

2.6 Arithmetic Coder.....22

2.7 The JPEG2000 Bit-stream.....24

2.7.1 Progression Order.....25

2.8 Markers and Marker Segments..... 28
2.9 Summary..... 33

3. Review of Channel Encoding Techniques 34
3.1 Channel Models..... 34
3.1.1 The Binary Symmetric Channel..... 35
3.1.2 Gilbert-Elliot Channel Model..... 37
3.1.3 The Rayleigh Fading Channel..... 39
3.2 Channel Coding Techniques..... 49
3.2.1 Cyclic Redundancy Check Codes..... 49
3.2.2 Convolutional Codes..... 52
3.2.2.1 The Transfer Function of a Convolutional Code..... 58
3.2.3 Decoding Convolutional Codes..... 61
3.2.3.1 The Viterbi Decoding Algorithm..... 62
3.2.4 Rate Compatible Convolutional Codes..... 64
3.2.5 Concatenation of CRC and RCPC Codes..... 65
3.2.6 Performance of RCPC Codes over Rayleigh Fading Channels... 67
3.3 Summary..... 70

4. Efficient Channel Protection for JPEG2000 images 71
4.1 Error-resilient Tools in JPEG2000..... 72
4.1.1 Data Partitioning and Resynchronization..... 72
4.1.2 Error Detection and Concealment..... 73
4.2 Hierarchical Structure of the JPEG2000 Bit-stream..... 75
4.3 Effect of Bit-errors on the JPEG2000 Bit-stream..... 79
4.3.1 Effect of Bit-errors in a Code-block..... 80
4.3.2 Effect of Bit-errors in a Packet..... 88

4.4 Equal Channel Protection 89
4.4.1 Overall Distortion of the Reconstructed Image..... 89
4.5 Summary..... 92

5. Unequal Channel Protection for JPEG2000 Images 93
5.1 Unequal Channel Protection across Layers..... 94
5.2 Unequal Channel Protection across Packets..... 99
5.3 Optimization of the Channel Protection..... 101
5.4 Simulation Results..... 103
5.5 Summary..... 132

6. Conclusions and Future Work 133

References 136

List of Tables

2.1 Le Gall 5/3 analysis and synthesis filter coefficients..... 16

2.2 Daubchines 9/7 analysis and synthesis filter coefficients..... 16

2.3 Example of the coding order of four quantized coefficients.....22

2.4 Markers and maker segments used in the JPEG2000 standard 30

3.1 Theoretical and obtained values for the fade durations of different fading channels.....46

4.1 Error resilient tools in JPEG2000..... 74

4.2 Bit-plane representation of quantized coefficients..... 83

4.3 Bit-plane representation of quantized coefficients after corruption..... 84

5.1 Code-block contributions to data packets.....96

5.2 Code-block inclusion information for the JPEG2000 compressed gray-level Lena image.....105

5.3 Puncturing matrices and RCPC codes.....106

5.4 Channel coding rates for the Lena gray-level image with channel protection by means of the ECP technique.....114

5.5 Protection schemes for the gray-level Lena image with channel

protection by means of the UCP-L technique at lossless compression.....115
5.6 Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.5 bpp.....116
5.7 Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.25 bpp.....117
5.8 Protection Schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.125 bpp.....118
5.9 Protection Scheme for the gray-level Lena image with channel protection by means of the UCP-P technique at lossless compression.....119
5.10 Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.5bpp.....120
5.11 Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.25 bpp.....121
5.12 Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.125 bpp.....122
5.13 Average PSNR and decoding probability for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh fading channel with a mobile speed of 3.6 *km/h*.....124
5.14 Average PSNR and decoding probability for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh fading channel with a mobile speed of 48.28 *km/h*.....125
5.15 Average PSNR and decoding probability for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh-fading channel with a mobile speed of 96.56 *km/h*.....126

List of Figures

1.1 Transmission environment for mobile receivers.....7

2.1 Block diagram of the JPEG2000 encoder.....12
2.2 Tiling process of the gray-level Peppers image into four tiles..... 13
2.3 Generic form for a one-dimension wavelet transform..... 14
2.4 Two-level dyadic decomposition of the gray-level Lena image using a 2D-DWT.....15
2.5 Partition of the sub-bands of a tile into packets, precincts and code-blocks..... 19
2.6 Code-block contributions for a JPEG2000 bit-stream with five layers and seven codeblocks..... 25
2.7 Example of SNR scalability on the gray-level Barbara at 0.125 bpp, 0.25 bpp and 0.5bpp.....27
2.8 Example of spatial scalability on the gray-level Barbara at three levels of progressive-by-resolution decoding..... 28
2.9 General structure of a marker segment in JPEG2000.....30
2.10 Organization of a JPEG2000 bit-stream for one tile and *n* packets.....32

3.1 Block diagram of a discrete channel model.....35

3.2 Transition process diagram for the binary symmetric channel..... 36
3.3 The Gilbert-Elliot channel model..... 37
3.4 Envelope of a received signal after transmission over a frequency non-selective, slow fading channel with a mobile speed of 3.6 *km/h* and a carrier frequency of 900 *MHz*..... 40
3.5 Realization of Jakes' model as an input-output channel model..... 45
3.6 Received fading envelope and burst of errors for a Rayleigh fading channel simulated by Jakes' model with $\overline{SNR} = 10\ dB$, $SNR_R = 2.155$ (FSK), normalized Doppler spread = $6.7200e^{-4}$ and signal level $R = -1\ dB$..... 47
3.7 Received fading envelope and burst of errors for a Rayleigh fading channel simulated by Jakes' model with $\overline{SNR} = 11\ dB$, $SNR_R = 2.155$ (FSK), normalized Doppler spread = $6.7200e^{-4}$ and signal level $R = -2\ dB$..... 47
3.8 Received fading envelope and burst of errors for a Rayleigh fading channel simulated by Jakes' model with $\overline{SNR} = 15\ dB$, $SNR_R = 2.155$ (FSK), normalized Doppler spread = $6.7200e^{-4}$ and signal level $R = -4\ dB$..... 48
3.9 Received fading envelope and burst of errors for a Rayleigh fading channel simulated by Jakes' model with $\overline{SNR} = 20\ dB$, $SNR_R = 2.155$ (FSK), normalized Doppler spread = $6.7200e^{-4}$ and signal level $R = -8\ dB$..... 48
3.10 Rate-½ convolutional encoder..... 53
3.11 Code tree for a rate-½ convolutional coder..... 55
3.12 Code trellis diagram for the code tree shown in Figure 3.11..... 56
3.13 Four-state diagram for the code tree shown in Figure 3.11..... 57
3.13 Modified state diagram for the convolutional coder shown in Figure 3.11..... 58
3.15 Modified state diagram for the convolutional coder shown in Figure 3.14..... 60

3.16 Depth-7 convolutional interleaver..... 63
3.17 Concatenation of a CRC code and a RCPC code..... 66

4.1 Resynchronization after a bit-error is found in an entropy coded bit-stream... 73
4.2 Organization of a JPEG2000 bit-stream for one tile and *n* packets.....76
4.3 Effect of bit-errors in the LL_L and *HH1* sub-band with *L=4* in the gray-level Baboon image encoded by means of the JPEG2000 standard with one qualitylayer.....77
4.4 Effect of bit-errors in the initial layer and final layer in the gray-level Baboon image encoded by means of the JPEG2000 standard with five quality layers.....78
4.5 Hierarchical structure of the JPEG2000 bit-stream for single and multiple-quality-layer compression..... 79
4.6 Effect of a single bit-error in the initial code-block coded data of a JPEG2000 bit-stream..... 81
4.7 A 3×3-sample code-block with nine quantized coefficients 83
4.8 A 3×3-sample code-block after corruption in the most significant bit plane and error concealment 86
4.9 Effect of single and multiple bit-errors in the initial code-block coded data when error concealment is used at the decoder.....87

5.1 64×64-sample code-blocks in the 512×512 gray-level Lena image with 3 levels of decomposition..... 95
5.2 Gray-level Lena image used for transmission simulations.....104
5.3 Probability of bit-error versus E_S / N_0 for a family of RCPC codes derived from the mother code of rate *R=1/4* and generator matrix *g*=[23 35 27 33].....107
5.4 Block diagram of the ECP coder.....108
5.5 Block diagram of the UCP-L coder.....109

5.6 Block diagram of the UCP-P coder.....109

5.7 Theoretical distortion of the reconstructed gray-level Lena image at lossless compression for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 *km*.....110

5.8 Theoretical distortion of the reconstructed gray-level Lena image at 0.5 and 0.25 bpp for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 *km/h*.....111

5.9 Theoretical distortion of the reconstructed gray-level Lena image at 0.125 bpp for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 *km/h*.....112

5.10 Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 10 dB for the gray-level Lena image.....127

5.11 Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 15 dB for the gray-level Lena image.....128

5.12 Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 20 dB for the gray-level Lena image.....129

5.13 Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 25 dB for the gray-level Lena image.....130

List of Abbreviations

1D-DWT	One-dimension discrete wavelet transform
2D-DWT	Two-dimension discrete wavelet transform
ARQ	Automatic repeat request
BER	Bit-error rate
BSC	Binary symmetric channel
CRC	Cyclic redundancy check
DMC	Discrete memoryless channel
EBCOT	Embedded block coding with optimized truncation
ECP	Equal channel protection
FEC	Forward error correction
FSK	Frequency-shift keying
GIF	Graphics interchange format
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunications Union
JPEG	Joint photography expert group
JPEG2000	Joint photography expert group version 2000
LPS	Least probable symbol
LSB	Least significant bit

LSBP	Least significant bit-plane
MMSE	Maximum mean square error
MPS	Most probable symbol
MSB	Most probable bit
MSE	Mean square error
MSPB	Most significant bit-plane
NCP	No channel protection
PPM	Packed packet-headers main-header
PPT	Packed packet-headers tile part-header
PSNR	Peak signal-to-noise ratio
QoS	Quality of service
RCPC	Rate compatible punctured convolutuonal
RMS	Root mean square
ROI	Region of interest
RVLC	Reversible variable length code
SNR	Signal-to-noise ratio
SOP	Start of packet
SPIHT	Set partitioning in hierarchical trees
TIFF	Tagged image file format
UCP	Unequal channel protection
UCP-L	Unequal channel protection across the layers
UCP-P	Unequal channel protection across the packets

Chapter 1

Introduction

The interest in communications of multimedia information, such as data, voice, images and video over wireless channels has increased considerably during the last few years. Wireless devices are expected to transmit large amounts of multimedia information, and efficient image transmission is one of the desired features.

An important characteristic of wireless channels is their limited bandwidths and high bit-error rates. Bandwidth constraints make necessary the use of compression techniques to reduce the amount of data to be transmitted. Image compression algorithms are designed to exploit the redundancy contained in images and produce shorter bit-streams powerful enough to reconstruct the image information with high quality. The resulting compressed bit-streams are highly susceptible to transmission errors, since the synchronization between the bit-stream and the decoder can be easily lost if bit-errors are present. Wireless channels cannot guarantee the necessary quality of service for compressed image data. Hence, in order to offer a good quality of service, error resilient transmission of compressed image data over error prone channels is crucial.

1.1 Error Resilience

An error resilient image communication system is a system with the ability to continue decoding the compressed image data even in the presence of errors in the bit-stream. The decoder should be able to produce a reconstructed image with a graceful degradation of image quality avoiding an irrecoverable failure even under poor channel conditions [1].

Error resilience can be achieved in three different ways: modifying the source coder, using a channel coding technique or developing a joint source-channel encoder [2]. The modification of the source coder involves the addition of redundant information to the bit-stream or the use of error resilient tools with a minimum addition of redundancy to the compressed data [3].

1.1.1 Error Resilient Tools

Error resilient tools are the mechanisms for providing a robust image compression system [1]. Typical error resilient tools include the use of markers in the bit-stream to prevent error propagation in variable-length codes. Several error resilient tools have been proposed recently in the literature [4-7]. Pettijohn et al. [4] have introduced a technique for error detection in arithmetic codes. The technique reserves a probability space for a symbol that is not in the source alphabet. This symbol is used at the decoder side to detect errors in the variable-length encoded data. The technique provides a significant improvement in the error resiliency of the bit-stream, with minimal rate overhead. Li et al. [5] have proposed the use of a bi-directional synchronization marker for image data compressed by means of a variable length coder. The proposed marker is able to recover the coding synchronization when a single or even two consecutive synchronization codes are corrupted.

Some other common tools used to reduce the susceptibility to bit-errors of compressed image data include the use of reversible variable-length coding (RVLC). RCVLC is capable of backward decoding and forward decoding. Hence, a RVLC needs at least two errors in the sequence to loss synchronization [8].

1.1.2 Channel Coding

Channel coding attempts to detect and correct bit-errors in the compressed image data without affecting the source coding process. Typical channel coding techniques include the use of forward error correction (FEC) techniques, automatic repeat request (ARQ) protocols or a hybrid FEC-ARQ scheme.

FEC techniques add controlled redundancy to the bit-stream to detect and correct possible bit-errors. This redundancy is usually achieved by means of block-codes, convolutional codes or a concatenation of both. If there is a knowledge of the channel conditions, the channel protection can be optimally assigned [9]. An inherent consequence of the use of FEC techniques is the increment on transmission rate, which is not always desirable if bandwidth constraints are present.

ARQ protocols are commonly employed over wireless or mobile channels to guarantee an error-free received image [10,11]. Here, if the decoder cannot understand the received information due to bit-errors or if the quality of the reconstructed image does not meet a specific requirement, a retransmission of the compressed image data is requested. One important disadvantage of ARQ protocols is the transmission delay introduced, which usually grows exponentially as the channel conditions deteriorate [1]. This delay can be reduced by the joint use of ARQ and FEC technique. In this case, the decoder demands a retransmission of the data only if the FEC technique is not capable to efficiently detect and correct the bit-

errors. Some important work involving hybrid ARQ-FEC techniques has been addressed in [12,13].

Another common way to improve the quality of the reconstructed images is the use of diversity techniques. Diversity techniques are generally employed when the channel conditions are significantly poor. Diversity involves the use of several replicas of the same information transmitted over independent channels to produce a higher quality reconstructed image. At the receiver side, fusion techniques [14] may be used to process the replicas of the compressed image data. Ramac et al. [15] have done some important work involving diversity techniques for image transmission and proposed a novel wavelet-domain diversity method for image transmission over wireless channels. The method is specially design for SPIHT [16] encoded images over fading channels and makes use of an unequal channel protection technique in conjunction with diversity to combat channel errors.

Some other approaches in the area of error resilience include the design of joint source-channel coding systems. This type of systems takes advantage of the information available about the image data, the source coding algorithm, the channel coding technique and the channel conditions to generate a compressed bit-stream resilient to errors. An important amount of research into joint source-channel coding schemes has been recently performed [17-25]. Even though this type of systems provides a good performance under noisy channels, the complexity of the coding process is high and involves modifying the source-coding algorithm.

1.2 Channel Protection for Images

According to Shannon's information separation theorem, source and channel coding can be separately optimized provided that the source and channel encoders

are allowed to operate on blocks of arbitrarily large length [26]. In addition, if the rate-distortion function of the encoded source is smaller than the channel capacity, the theoretically feasible performance is restricted exclusively by source coding errors [27]. Shannon's information separation theorem can be exploited to design a channel protection system without affecting the source-coding algorithm. However, even when the compressed image bit-stream is almost statistically independent, the bits may have different sensitivities. Hence, these bits should be protected according to their importance in the reconstructed image.

Some important work has been done to address this issue. Man et al. [28] have proposed a novel wavelet/sub-band image coder to produce a bit-stream that contains two different bit sequences. These bit sequences are channel protected according to their importance and channel noise sensitivity levels. The result is a performance improvement on the error resilience of the bit-stream. An unequal channel protection technique for progressive image transmission was proposed by Sherwood et al. [29]. In this technique, a SPIHT encoded image is packetized and protected by means of a concatenation of a block code and a convolutional code. The channel protection is assigned so to keep the progressive property of the image coder. A similar approach was used by Cosman et al. [9] for zerotree wavelet encoded images [30]. Li et al. [5] have introduced a layered bit-plane error protection technique for SPITH encode images. This technique assigns the channel protection to different bit-planes according to their significance in the reconstructed image. The technique uses an unequal channel protection scheme.

1.3 Motivation

Most of the work done in the area of channel protection for compressed image data uses specific encoders that do not follow any international standard. The majority of

images in the Internet, digital libraries and databases are compressed by means of coding systems that follow an international standard, such as GIF, JPEG, TIFF and PNG [31]. The use of standards facilitates the distribution and access to the information.

One of the most popular image compression standards is JPEG (Joint Photography Expert Group), jointly developed and established by the International Organization for Standardization (ISO), the International Telecommunications Union (ITU) and the International Electrotechnical Commission (IEC). Recently, these international organizations have already finished the first draft of a new version of the JPEG standard, called JPEG2000 [32].

The emerging JPEG2000 standard has been designed to address the limitations of the previous JPEG standard. Among some other important features, the new JPEG2000 standard has been designed to provide a better resiliency to errors compared to the current standard. This has been achieved by introducing a set of error resilient tools.

The growing popularity of the JPEG2000 standard and the increasing interest in multimedia communications are the main motivation to study the problem of robust image transmission over noisy channels.

1.4 Major Contributions

In this thesis, the issue of robust transmission of JPEG2000 images over personal mobile communication channels (Rayleigh fading channels) is addressed. Fig. 1.1 shows a block diagram of the transmission environment assumed in this work.

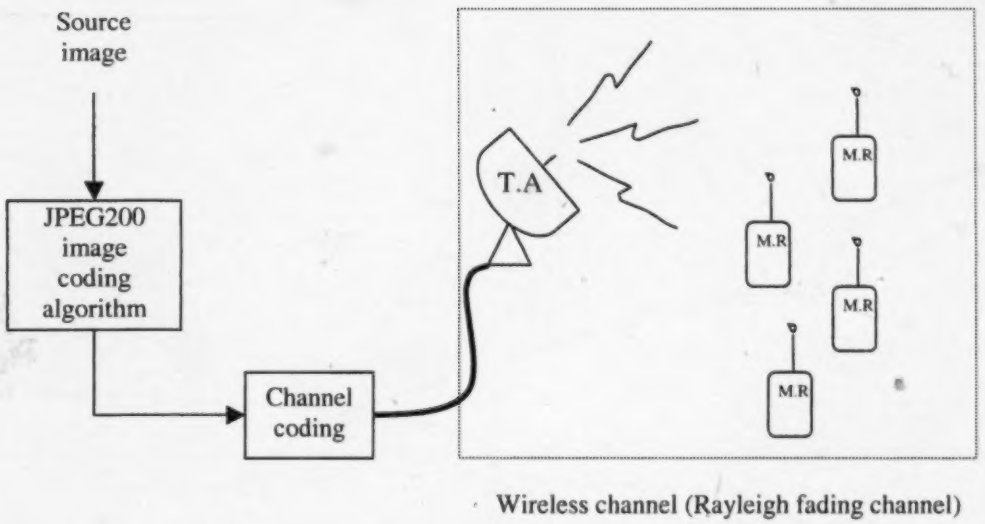


Figure 1.1. Transmission environment for mobile receivers. T.A: transmission antenna. M.R: mobile receiver

The major contributions of this thesis are in the area of channel coding. Two novel unequal channel protection techniques are proposed:

i) Unequal channel protection across the layers [33]. The technique protects the different layers of the JPEG2000 bit-stream unequally according to the energy contained in each layer.

ii) Unequal channel protection across the packets [34]. Within a layer, the technique unequally assigns channel protection to the packets according to their energy.

The proposed techniques take advantage of the error resilient tools included in the JPEG2000 and assign the channel protection according to the current channel conditions. The channel protection is achieved by means of a cyclic redundancy check outer coder and an inner rate-compatible convolutional coder. Since the proposed techniques employ a FEC scheme to improve the resilience to errors of the JPEG2000 bit-stream, the source coder is not modified. Simulation results show that the proposed techniques improve the error resiliency of a JPEG2000 compression system over wireless channels.

1.4 Organization of the Thesis

The thesis is organized as follows. Chapter 2 provides a review of the JPEG2000 standard, emphasizing some of its most important features, including layered compression and the bit-stream formation. The channel model used to simulate Rayleigh fading channels is described in chapter 3. Chapter 3 also discusses the theory behind the two FEC techniques used in this work: cyclic redundancy check codes and rate-compatible convolutional codes. An analysis of the performance of rate-compatible convolutional codes over Rayleigh fading channels is described in chapter 3 as well. This analysis is useful in the design of the unequal channel protection techniques proposed in this thesis.

The error resilient tools adopted in the JPEG2000 standard are discussed in chapter 4 followed by an analysis of the effect of bit-errors in the JPEG2000 bit-stream. This analysis is used to design the unequal channel protection techniques. Chapter 4 also describes an equal channel protection technique for JPEG2000 images.

The unequal channel protection techniques proposed in this thesis are discussed in chapter 5. Simulations are performed using a Rayleigh fading channel with different

channel conditions and results from the unequal channel protection techniques are compared to the performance of the equal channel protection technique and the case of no channel protection. Chapter 6 gives the conclusion about the work presented in this thesis, followed by some suggestions for future work.

Chapter 2

Review of the JPEG2000 Image Coding Standard

The new JPEG2000 standard for still images is intended to overcome the shortcomings of the existing JPEG standard. The JPEG standard has been in use for almost a decade now and although it provides a good compression performance, it does not fulfill advanced requirements such as scalability and interoperability in network and mobile environments [35].

The standardization process, coordinated by the Joint Technical Committee on Information technology of the ISO/IEC [32] has already finalized the Final Draft International Standard Part I. In this chapter, a review of the JPEG2000 standard, according to Part I of the standardization process is presented.

JPEG2000 makes use of the wavelet and sub-band technologies, and incorporates functionalities such as lossless and lossy compression, spatial and quality scalability, region of interest coding, random code-stream access and error-resilient coding [32]. Some of the markets targeted by the JPEG2000 standard are Internet, printing, digital photography, remote sensing, mobile, digital libraries and E-commerce. The most important features of the JPEG2000 standard are described in the following [35]:

i) Lossless and lossy compression: the standard provides lossy compression with a superior performance to the current standard at low bit-rates (e.g. below 0.25 bpp for highly detailed gray-level images). It also provides lossless compression with progressive decoding. Applications such as digital libraries/databases and medical imagery can benefit from this feature.

ii) Protective image security: the open architecture of the JPEG2000 standard makes easy the use of protection techniques of digital images such as watermarking, labeling, stamping or encryption [35].

iii) Region-of-interest coding: in this mode, regions of interest (ROI's) can be defined. These ROI's can be encoded and transmitted with better quality than the rest of the image [32].

iv) Robustness to bit errors: the standard incorporate a set of error resilient tools to make the bit-stream more robust to transmission errors. These tools are described in detail in Chapter 4.

2.1 The JPEG2000 Standard Algorithm

The core compression algorithm is primarily based on the Embedded Block Coding with Optimized Truncation (EBCOT) of the bit-stream [36]. The EBCOT algorithm provides a superior compression performance and produces a bit-stream with features such as resolution and SNR scalability and random access. Figure 2.1 shows the block schematic of the JPEG2000 compression engine.

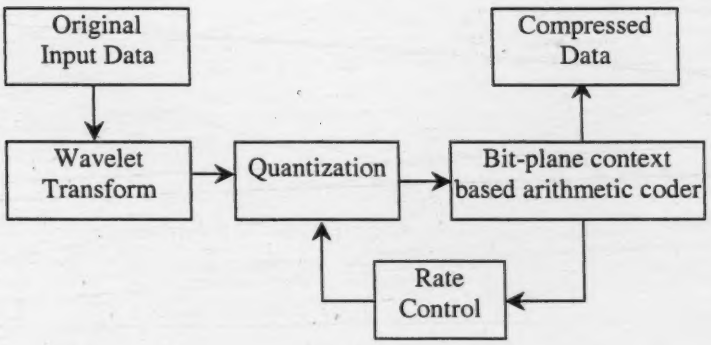


Figure 2.1 Block diagram of the JPEG2000 encoder.

In JPEG2000, the image to be compressed is first DC level shifted by subtracting the component depth. DC level shifting is performed on components that are unsigned only. When color transformation is used, DC level shifting is performed before computation of the forward component transform.

Once the image has been DC level shifted, it is divided into rectangular non-overlapping blocks called tiles. This first process is optional and the entire image can be regarded as one tile. Each tile is treated as a different image and compressed independently. Fig. 2.2 shows the tiling process on a gray-level image with one component.

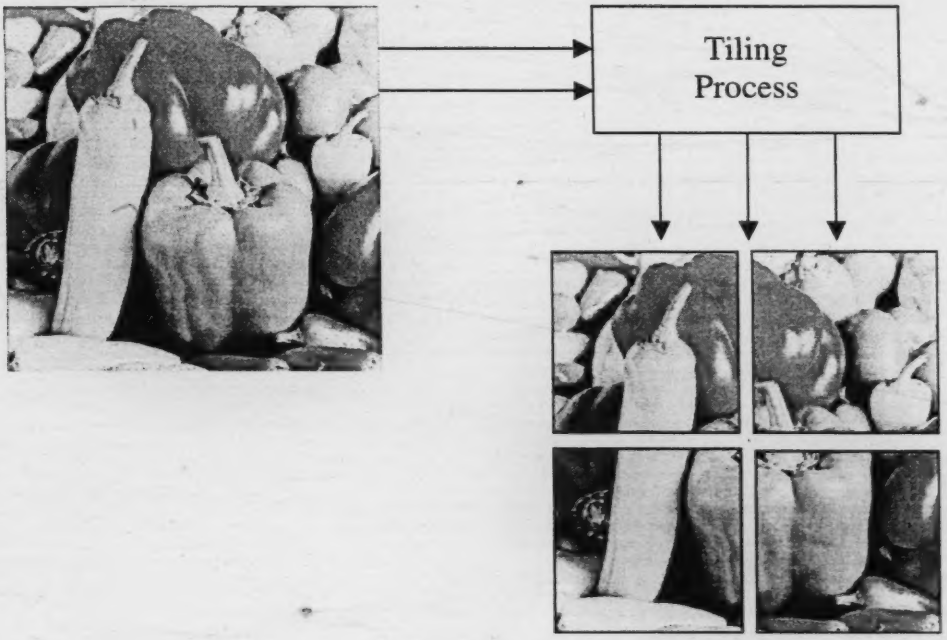


Figure 2.2. Segmentation of the gray-level *Peppers* image into four tiles.

The use of a tiling process reduces memory requirements, and because each tile is encoded independently, they can be used to decode specific parts of the image instead of the whole image.

2.2 Discrete Wavelet Transform

The tile components are decomposed into different decomposition levels using a two-dimensional discrete wavelet transform (2D-DWT). The 2D-DWT is realized as an extension of the one-dimensional discrete wavelet transform (1-D DWT) using separable wavelet filters [32]. The generic form for a one-dimension transform is shown in Fig. 2.3.

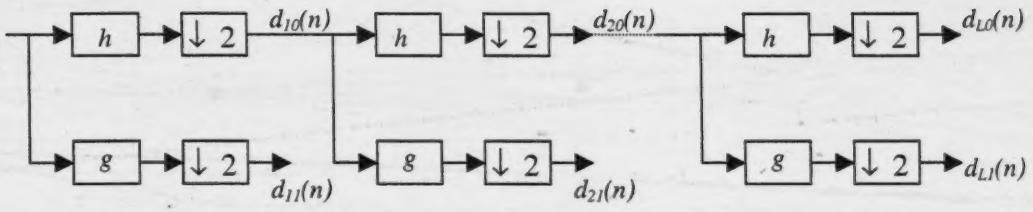


Figure 2.3. Generic form for a one-dimension wavelet transform.

In the 1-D wavelet transform, a one-dimensional signal is passed through a set of lowpass and highpass filters, h and g respectively, and down sampled by a factor of two. This process constitutes one level of transform. Multiple decomposition levels are made by repeating the filtering and decimation process on the lowpass filter output only. The process can be carried out for a finite number of levels L and the resulting coefficients $d_{i1}(n), i \in \{1,....L\}$ and $d_{L0}(n)$ are called the transform coefficients.

Applying a 1-D transform to all the rows of the input, and then repeating on all the columns compute the 2-D transform. Each time the 2-D wavelet transform is applied, a level of decomposition is achieved. A decomposition level contains a number of sub-bands that describe the horizontal, vertical and diagonal spatial frequency characteristics of the original tile component. Part I of the standard allows only power of two decompositions in the form of dyadic decomposition. Fig. 2.4 shows an example of a two-level dyadic decomposition with seven sub-bands of the gray-level Lena image. In this example, $L=2$ represents the highest level of decomposition.

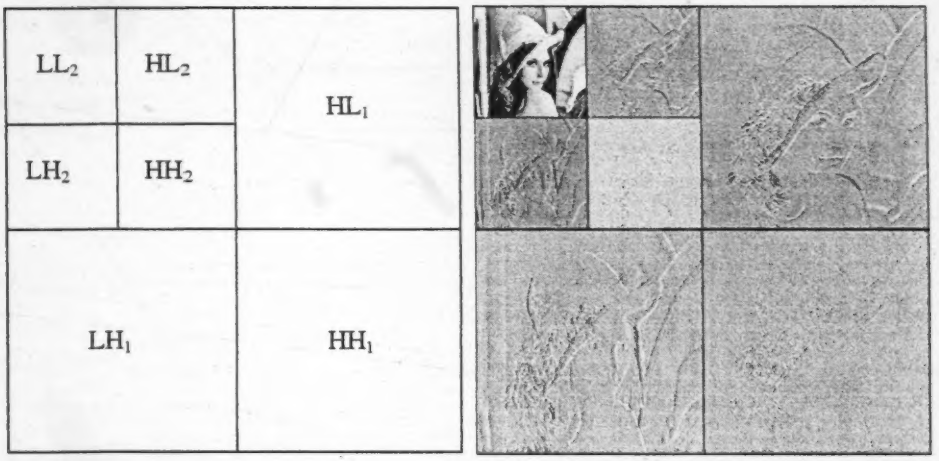


Figure 2.4. A two-level (*L=2*) dyadic decomposition of the gray-level Lena image using a 2D-DWT. Sub-bands *HL* contain the horizontal frequency characteristics of the image. Sub-bands *LH* contain the vertical frequency characteristics of the image. Sub-bands *HH* contain the diagonal frequency characteristics of the image.

In order to perform the inverse discrete wavelet transformation (IDWT), the JPEG2000 standard uses a one-dimensional sub-band recomposition of a one-dimensional set of samples from low-pass and high-pass coefficients.

In JPEG2000, the DWT can be reversible or irreversible. The default irreversible transform is implemented by means of the Daubchines 9-tap/7-tap filter. The default reversible transformation is implemented by means of the Le Gall 5-tap/3-tap filter. The analysis and synthesis coefficients of the default reversible and irreversible DWT are given in Tables 2.1 and 2.2.

Table 2.1. Le Gall 5/3 analysis and synthesis filter coefficients [32]

Analysis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	6/8	1
+/- 1	2/8	-1/2
+/- 2	-1/8	
Synthesis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	1	6/8
+/- 1	1/2	-2/8
+/- 2		-1/8

Table 2.2. Daubchines 9/7 analysis and synthesis filter coefficients [32]

Analysis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	0.6029490182363579	1.115087052456994
+/- 1	0.2668641184428723	-0.5912717631142470
+/- 2	-0.07822326652898785	-0.05754352622849957
+/- 3	-0.01686411844287495	0.09127176311424948
+/- 4	0.02674875741080976	
Synthesis Filter Coefficients		
i	Low-Pass Filter	High-Pass Filter
0	1.115087052456994	0.6029490182363579
+/- 1	0.5912717631142470	-0.2668641184428723
+/- 2	-0.05754352622849957	-0.07822326652898785
+/- 3	-0.09127176311424948	0.01686411844287495
+/- 4		0.02674875741080976

The default wavelet transforms in JPEG2000 are biorthogonal wavelets; that is, the wavelets have some orthogonality relations between their filters. An orthogonal wavelet has wavelet filters that satisfy the orthogonality constraints [37]:

$$\sum_n h(n-2i)h(n-2j) = \delta(i-j) \tag{2.1}$$

$$\sum_n g(n-2i)g(n-2j) = \delta(i-j) \tag{2.2}$$

$$\sum_n h(n-2i)g(n-2j) = 0 \tag{2.3}$$

When the conditions in Eqs. (2.1-2.3) are met, the wavelet transform can be viewed as projecting the input signal onto a set of orthogonal basis functions. If the filters are normalized, as in Eqs. (2.1-2.3), the resulting wavelet transform is energy preserving. This energy conservation property, analogous to the Parseval theorem in Fourier analysis, is defined as:

$$\sum_{n=0}^{N-1} x^2(n) = \sum_{l=0}^{L-1} w^2(l) \tag{2.4}$$

This energy preservation property is useful in the design of coding and compressing systems since the mean square distortion introduce in the transform coefficients equals the mean square distortion in the reconstructed signal.

Biorthogonal wavelets differ from orthogonal wavelets in that the forward wavelet transform is equivalent to projecting the input signal onto non-orthogonal basis functions. Hence, biorthogonal wavelets are not energy preserving.

2.3 Quantization

After the DWT, all coefficients are quantized using a scalar quantization (Part I). This process is lossy, unless the quantization step is one and the coefficients are integers, as produced by the reversible 5-tap/3-tap filter. A transform coefficient $a_b(u,v)$ of the sub-band b is quantized to the value $q_b(u,v)$ according to the equation:

$$q_b(u,v) = sign(a_b(u,v))\left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor \tag{2.5}$$

The quantization step Δ_b is calculated using the dynamic range R_b of sub-band b, the exponent ε_b and mantissa μ_b as:

$$\Delta_b = 2^{R_b-\varepsilon_b}\left(1+\frac{\mu_b}{2^{11}}\right) \tag{2.6}$$

The dynamic range R_b depends on the number of bits used to represent the original image tile component and on the choice of wavelet transform. Note that all quantized transform coefficients are signed. These coefficients are expressed in a sign-magnitude representation prior coding.

2.4 Precincts, Layers and Packets

After quantization, the sub-bands are divided into non-overlapping rectangles called precincts. Three spatially consistent precincts, one from each sub-band at the same resolution level, comprise a packet. Each precinct is further divided into smaller non-overlapping rectangles called code-blocks. The data in a packet is

ordered in such a way that the contribution from the *LL, HL, LH*, and *HH* sub-bands appear in that order. Fig. 2.5 shows the partition of the sub-bands of a tile into packets, precincts and code-blocks. Here, the image is decomposed with three levels of decomposition. The code-block size is equal to the size of the *LL3* sub-band and each precinct contains four code-blocks, except for the *L=3* resolution level, where a precinct contains a single code-block. With this configuration, six packets comprise the final bit-stream. The first packet contains all code-blocks in all sub-bands at the *L=3* resolution level, the second packet is comprised by all code-blocks in all sub-bands at the *L=2* resolution level, while the last four packets contain all code-blocks in all sub-bands at the first resolution level.

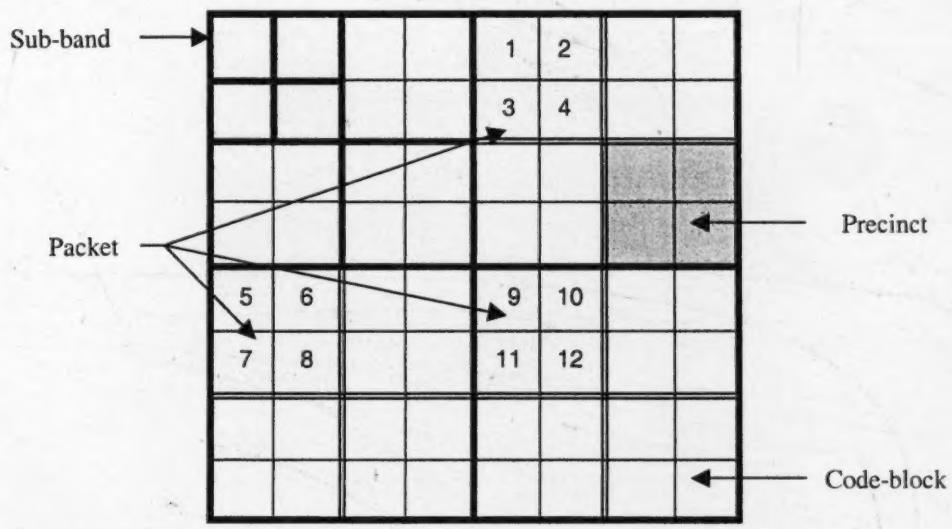


Figure 2.5. Partition of the sub-bands of a tile into packets, precincts and code-blocks.

2.5 Bit-stream Formation

The coefficients contained in the code-blocks are expressed in a bit-plane representation. These individual bit-planes are encoded one at a time starting with the most significant bit-plane with a nonzero element to the least significant bit-plane [32]. The individual bit-planes are coded within three coding passes. Each coefficient bit in a bit-plane is coded in only one of the three coding passes. The coding passes are called significance propagation, magnitude refinement, and cleanup [32]. For each pass, contexts are created which are then provided to the arithmetic coder along with the bit-stream to produce the coded data for a code-block. During the coding process, the sub-bands are visited in raster order starting with the LL_L sub-band. Within a packet, the code-blocks are visited in raster order as well.

The first coding pass for each bit plane is the significance pass. This pass is used to obtain significance and sign information for samples that have not yet been found to be significant and are predicted to become significant during the processing of the current bit plane. If a sample has not yet been found to be significant, and is predicted to become significant, the significance of the sample is coded with a single bit. If the sample also happens to be significant, its sign is coded using a single bit as well. If the most significant bit plane is being processed, all samples are predicted to remain insignificant. Otherwise, a sample is predicted to become significant if any of the 8-connected neighbors has already been found to be significant. Therefore, the significance and refinement passes for the most significant bit plane are not coded.

The bit used to represent the significance information is coded using one of nine contexts during the arithmetic coding process. The particular context used is selected based on the significance of the eight surrounding neighbor coefficients of the current sample and the orientation of the sub-band where the sample is located (e.g., *LL*, *LH*, *HL*, *HH*). On the other hand, the sign of a sample is coded as the difference

between the actual and predicted sign. Sign prediction is performed using the significance and sign information for the four-connected neighbor coefficients of the current sample.

The second coding pass for each bit plane is the refinement pass. This pass encodes subsequent bits after the most significant bit for each sample. If a sample was found to be significant in a previous bit plane, the next most significant bit of that sample is coded using a single bit. Each refinement bit is coded using one of three contexts during the arithmetic coding process. The particular context employed is selected based on if the second most significant bit position is being refined and the significance of the eight surrounding neighbor coefficients of the current sample.

The final coding pass for each bit plane is the cleanup pass. This pass is used to encode significance and sign information for those samples that have not yet been found to be significant and are predicted to remain insignificant during the processing of the current bit plane. The cleanup pass is similar to the significance pass. However, the main difference is that the cleanup pass encodes information about samples that are predicted to remain insignificant, rather than those that are predicted to become significant. The cleanup pass not only uses the neighbor context, like that of the significance propagation pass, but also a run-length context.

It is important to notice that within each sub-band, the code-blocks are encoded independently, confined by the boundaries established by the corresponding precinct. Table 2.3 shows an example of the coding order for four quantized coefficients. In this example, all neighbors not included in the table are assumed to be identically zero. The table indicates in which pass each bit is coded.

Table 2.3. Example of the coding order of four quantized coefficients [32].

Coding Pass	Coefficient value			
	10	1	3	-7
Clean-up	1+	0	0	0
Significance		0		
Refinement	0			
Clean-up			0	1-
Significance		0	1+	
Refinement	1			1
Clean-up				
Significance		1+		
Refinement	0		1	1
Clean-up				

The sign bit is coded after the initial 1 bit and is indicated in the table by the + or - sign. The first pass in a new set of coefficients is always a clean-up pass because there can be no predicted significant, or refinement bits. After the first pass, the decoded 1 bit of the first coefficient causes the second coefficient to be coded in the significance pass for the next bit-plane. The 1 bit coded for the last coefficient in the second clean-up pass causes the third coefficient to be coded in the next significance pass.

2.6 Arithmetic Coder

As mentioned before, the contexts created for each bit-plane of a code-block are provided to the arithmetic coder along with the bit-stream to produce the coded data for a code-block. The arithmetic coder adopted in the JPEG2000 standard is a

context dependent binary arithmetic coder. The compression is performed relative to an adaptive probability model associated with each of 18 different coding contexts. The context models are always reinitialized at the beginning of each code-block and the arithmetic coder is always terminated at the end of a code-block.

The recursive probability interval subdivision of Elias coding is the basis for the binary arithmetic coding process. With each binary decision the current probability interval is subdivided into two sub-intervals, and the code string is modified (if necessary) so that it points to the base (the lower bound) of the probability sub-interval assigned to the symbol that occurred.

In the partitioning of the current interval into two sub-intervals, the sub-interval for the more probable symbol (MPS) is ordered above the sub-interval for the less probable symbol (LPS). Therefore, when the MPS is coded, the LPS subinterval is added to the code string. This coding convention requires that symbols be recognized as either MPS or LPS, rather than 0 or 1. Consequently, the size of the LPS interval and the sense of the MPS for each decision must be known in order to code that decision.

Since the code string always points to the base of the current interval, the decoding process is a matter of determining, for each decision, which sub-interval is pointed to by the compressed data. This is also done recursively, using the same interval sub-division process as in the encoder. Each time a decision is decoded, the decoder subtracts any interval the encoder added to the code string. Therefore, the code string in the decoder is a pointer into the current interval relative to the base of the current interval. Since the coding process involves addition of binary fractions rather than concatenation of integer code words, the more probable binary decisions can often be coded at a rate of much less than one bit per decision.

2.7 The JPEG2000 Bit-stream

The individual code-streams obtained by compressing each code-block are the basic unit of the final JPEG2000 bit-stream. These individual code-streams have the property that they can be truncated to a variety of discrete lengths, with an associated distortion when reconstructing from each of these truncated code-streams.

Once the image has been compressed, the encoder determines the extent to which each individual code-stream should be truncated to achieve a target compression ratio or distortion. This is achieved by means of a post-processing operation that passes all the compressed code-block code-streams.

The JPEG2000 bit-stream is composed from a collection of layers, where each layer is a sequential contribution to the overall image quality. Each layer contains the additional contributions from each code-block. The first, lowest quality layer is formed from the optimally truncated code-block bit-streams. Each subsequent layer is formed by optimally truncating the code-block bit-streams to achieve successively higher target bit-rates, distortion bounds or other quality metrics [32]. Fig. 2.6 shows the code-block contributions for a JPEG2000 bit-stream with five layers and seven code-blocks. Note that some code-block contributions may be empty.

The code-block truncation points associated with each layer are optimal in the rate-distortion sense, which means that the bit-stream obtained by discarding a whole number of least important layers will always be rate-distortion optimal. In order to achieve total optimality in the rate-distortion sense, the bit-stream should be truncated at a layer point. However, by using several layers, rate-distortion optimality can still be achieved even if the bit-stream is truncated part way through a layer.

The final JPEG2000 bit-stream is then comprised by a succession of layers and a set of marker segments that contain all the necessary information to decode the information represented in the layers. The marker segments will be described in detail is section 2.8.

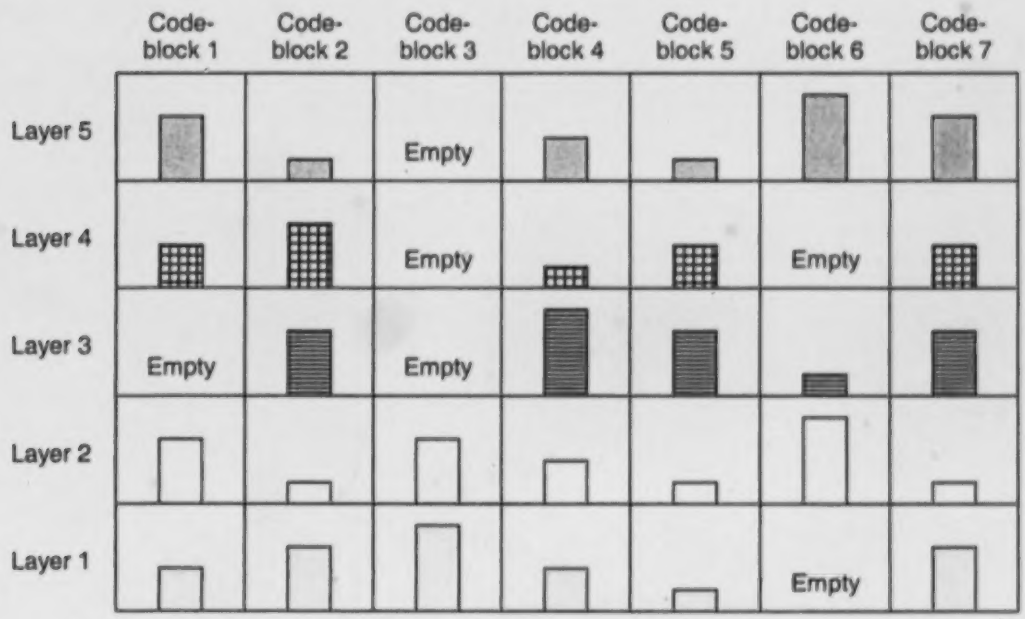


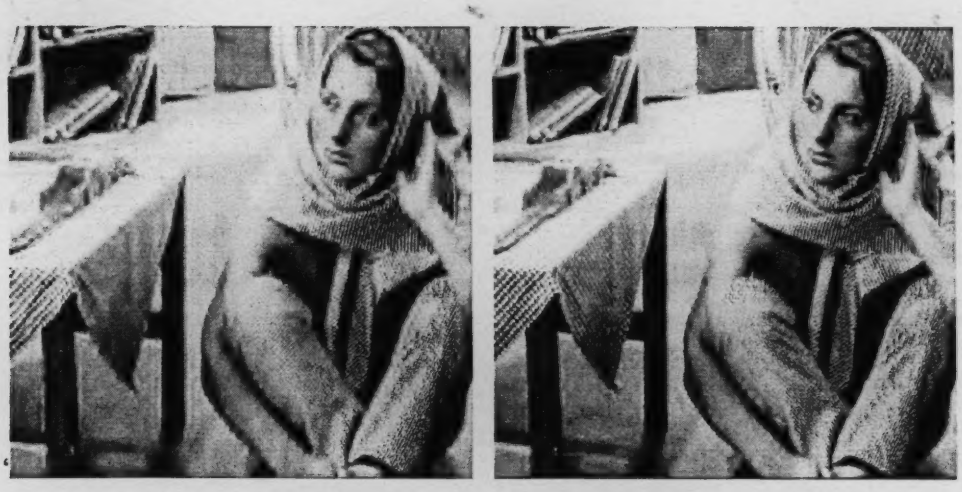
Figure 2.6. Code-block contributions for a JPEG2000 bit-stream with five layers and seven code-blocks [32].

2.7.1 Progression Order

The packets in the final JPEG2000 code stream contain information from a specific tile, component, layer, resolution and precinct. The order in which these packets are interleaved is called the progression order. The interleaving of the packets can progress along four axes: layer, component, resolution and precinct.

Progression by layer-resolution-component-precinct, or SNR scalability, is achieved when a minimum of two quality layers of the same spatial resolution are used to compress the image information [35]. The lower layer, or initial layer, is encoded to provide the basic image quality and the enhancement layers are coded to enhance the initial layer. Because the decoder is able to decode the quality layers in sequence, each enhancement layer, when added to the initial layer, produces a higher quality reconstructed image. One important advantage of the SNR scalability is the resiliency to errors it provides, as the initial layer can be sent over a channel with a good error performance, while the enhancement layers can be sent over a channel with poor error performance. Fig. 2.7 shows an example of SNR scalability. The image is compressed using a lossless scheme and decompressed at 0.125 bpp, 0.25 bpp and 0.5 bpp.

Progression by resolution-layer-component-precinct, or spatial scalability, is achieved when a minimum of two layers of spatial resolution are used to compress the image information [35]. The initial layer, in this case, is coded to provide the basic spatial resolution and the enhancement layers employ the spatially interpolated initial layer and achieve the full spatial resolution of the image. Spatial scalability is useful for database access and in communications where the receivers have different capabilities in terms of display and bandwidth. Figure 2.8 shows an example of spatial scalability with three levels of progressive-by-resolution decoding.



(a) (b)

(c)

Figure 2.7. SNR scalability on a gray-level image. The *Barbara* image is decompressed at (a) 0.125 bpp, (b) 0.25 bpp and (c) 0.5 bpp.

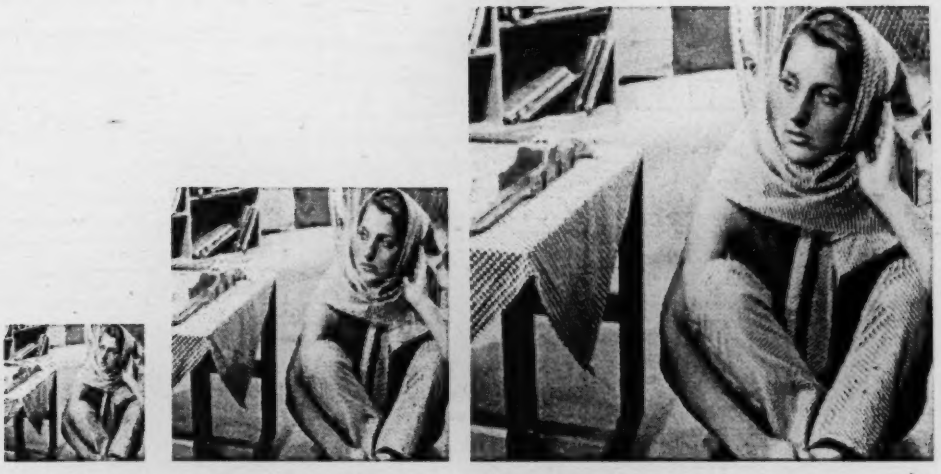


Figure 2.8. Spatial scalability on a gray-level image. The *Barbara* image is reconstructed at three levels of progressive-by-resolution decoding.

2.8 Markers and Marker Segments

The JPEG2000 standard makes use of markers and marker segments to delimit and signal the characteristics of the bit-stream. The markers and marker segments are grouped in the headers. There are two types of headers:

i) The main header is found at the beginning of the code stream.

ii) The tile-part headers are found at the beginning of each tile-part

Some markers and marker segments are restricted to only one of the two types of headers while others can be found in either.

Every marker is two bytes long. The first byte consists of a single 0xFF byte. The second byte denotes the specific marker and can have any value in the range 0x01 to 0xFE. A marker segment includes a marker and associated parameters, called marker

parameters. In every marker segment the first two bytes after the marker shall be an unsigned big endian integer value that denotes the length in bytes of the marker parameters, including two bytes of this length parameter but not the two bytes of the marker itself.

There are six types of markers in the JPEG2000 standard:

i) Delimiting.
ii) Fixed information.
iii) Functional.
iv) In bit-stream.
v) Pointer.
vi) Informational.

Delimiting markers and marker segments are used to frame the headers and the data. Fixed information marker segments contain information about an image. Functional marker segments are used to describe the coding functions used. In bit-stream markers and marker segments are used for error resilience. Pointer marker segments point to specific offsets in the bit stream. Informational marker segments provide supplementary information.

The general structure of a marker segment is depicted in Fig 2.9. The marker segments are designated by a three-letter abbreviation. The parameter values have capital letter designations with the marker's abbreviation as a subscript. A rectangle is used to indicate the parameters in the marker segment. The width of the rectangle is proportional to the number of bytes in the field. A rectangle with diagonal stripes indicates that the parameter is of varying size. Two parameters with superscripts and a gray area between indicate a run of several of these parameters.

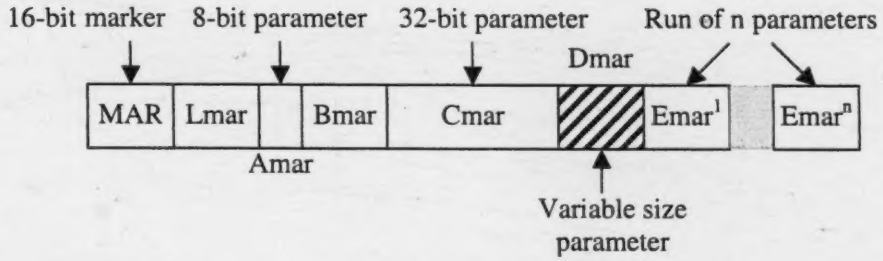


Figure 2.9. General structure of a marker segment in JPEG2000 [32].

Table 2.4 summarizes some of the markers and maker segments used in the JPEG2000 standard by category. Note that not all markers and marker segments need to appear in the final bit-stream. Its usage depends on the features used to encode the image information.

Table 2.4. Markers and maker segments used in the JPEG2000 standard by category.

Information	Marker segment name
Delimiting marker segments	
Start of codestream	SOC
Start of tile-part	SOT
Start of data	SOD
End of codestream	EOC
Fixed information marker segments	
Tile size (height and width)	SIZ
Number of components	
Component transform used	
Component precision	
Component mapping to the reference grid (sub-sampling)	

Functional marker segments	
Coding style Number of decomposition levels Progression order Number of layers Code-block size Code-block style Wavelet transform COD, COC	COD COC
No quantization Quantization	QCD QCC
Progression starting point Progression ending point Progression order default	POD
Pointer marker segments	
Tile-part data length	TLM
Packet lengths	PLM, PLT
Code-block values for new layers Code-block layer number Code-block inclusion Maximum bit depth Truncation point Bit stream length for decomposition level and layer in a code-block	Packet header,PPM, PPT
In stream marker segments	
Error resilience End of packet header	SOP, EPH

Information marker segment	
Optional information	CME

The final structure of the JPEG2000 bit-stream is depicted in Fig. 2.10. It is assumed that the image has been compressed using one tile with one component and *n* packets.

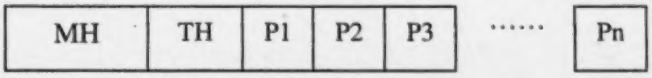


Figure 2.10. Organization of a JPEG2000 bit-stream for one tile and *n* packets. MH: Main Header. TH: Tile-part Header. Pn: *nth* Packet.

The Main Header consists of the Start of Code Stream marker (SOC), the Image and Tile Size (SIZ) marker segment, the Coding Style Default (COD) marker segment, the Quantization Default (QCD) marker segment and the Packed Packet headers-Main header (PPM) marker segment. The PPM marker segment is the collection of all the packet headers.

The Tile-part Header consists of the Start of Tile-part (SOT) marker and Start of Data (SOD) marker. The last marker is always the End of Code-stream (EOC) marker. The data packets are found between the SOC marker and the EOC marker, if only one tile is used, or between a SOC marker and a SOT marker, if multiple tiles are used. A Start of packet (SOP) marker may be placed in front of every packet for error resiliency purposes.

2.9 Summary

A review of the JPEG2000 standard for still images was presented in this Chapter. The most important features of the standard were discussed, including the arithmetic coding process, the bit-stream formation and layered compression. Specifically, the organization of the JPEG2000 code-stream into different quality layers was described emphasizing the importance of code-block contributions to the different layers. The structure of the final JPEG2000 bit-stream was described as well. In particular, the use of markers and marker segments in the JPEG2000 standard to delimit and signal the characteristics of the bit-stream was discussed. In addition, the importance of the information contained in the marker segments was described.

Chapter 3

Review of Channel Encoding Techniques

In this Chapter, a review of some channel models and coding techniques is presented. Specifically, the main characteristics and properties of cyclic redundancy check and rate compatible convolutional codes are analyzed. This analysis is useful in the design of the channel protection techniques proposed in this work.

3.1 Channel Models

Multimedia information is likely to go through different communication environments. A model of a communication environment is a mathematical representation that describes the way signals are affected by interference, amplitude and phase fluctuations, noise and equipment flaws. A channel model is commonly defined including the modulator, demodulator and all intermediate transmission equipment. Fig. 3.1 shows a block diagram of a channel model defined by the modulator input, the demodulator output and the statistics

describing the relation between the inputs and possible outputs. This type of channel model is commonly called a discrete channel model [38]. In the following, three different channel models commonly used to analyze and simulate digital communications systems are described. It is assumed that hard-decision modulation is performed in these channel models. Hard-decision decoding is realized when the modulator input x has a value 0 or 1 and the demodulator output y has a value 0 or 1.

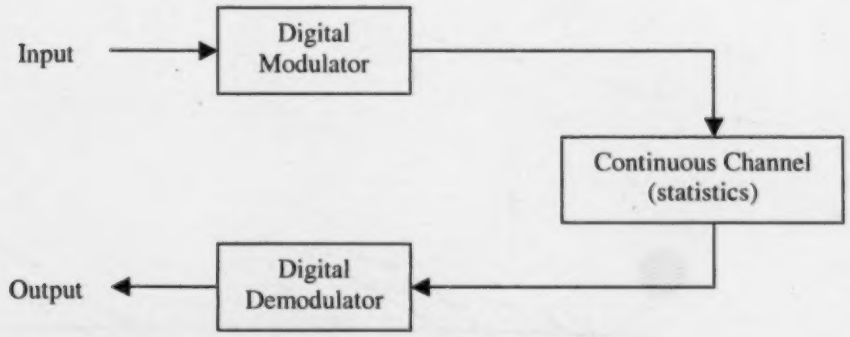


Figure 3.1. Block diagram of a discrete channel model.

3.1.1. The Binary Symmetric Channel

The Binary Symmetric Channel (BSC) is the simplest type of Discrete Memoryless Channel (DMC) model. A channel is said to be memoryless because the output at any time only depends statistically on the input at that time. The BSC is commonly used to describe any real independent-error channel, such as those channels affected by atmospheric impulse noise or intersymbol interference.

Let a DMC model have an input x, which can take a value of 1 or 0, and an output y, which can take a value of 1 or 0 as well. Let us now suppose that for every transmitted bit x and regardless of previous transmitted or received information, we receive x in error with probability of p or correctly with probability of $1\text{-}p$. This situation can be expressed using conditional probabilities as follows [38]:

$$P(y=1|x=0) = P(y=0|x=1) = p \tag{3.1}$$

$$P(y=0|x=0) = P(y=1|x=1) = 1-p \tag{3.2}$$

Equations (3.1) and (3.2) describe a Binary Symmetric Channel. Fig. 3.2 depicts the transition diagram process for the BSC. The bit error rate (BER) for a BSC is given by the probability of error p.

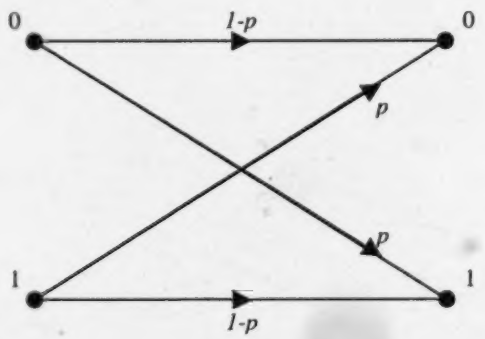
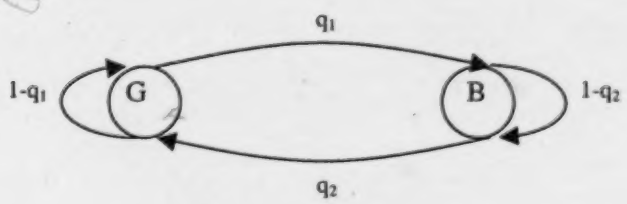


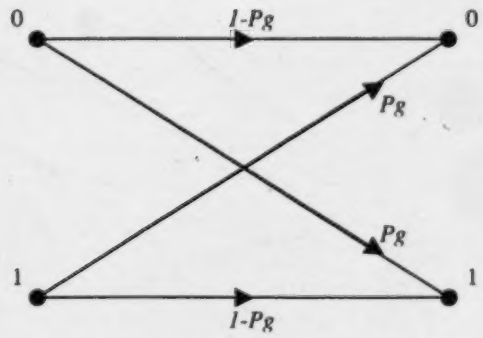
Figure 3.2. Transition process diagram for the binary symmetric channel.

3.1.2 Gilbert-Elliot Channel Model

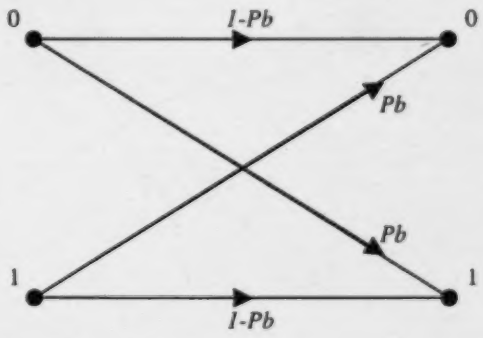
The BSC model provides a good approximation to channels with no memory; however, it fails to describe channels with fades, which arise in wireless communications. A fading channel is mainly characterized by its memory. The output now depends statistically on previous sent and received information. Fig. 3.3(a) shows a state diagram for a two-state Markov model. This model was first used by Gilbert [39] to characterize the error sequences and memory associated with fading channels.



(a) A two-state Markov model describing a fading channel



(b) Good state (G)



(c) Bad state (B)

Figure 3.3. The Gilbert-Elliot channel model.

In each state, the channel behaves as a BSC with a certain probability of error, *Pg* for the good state and *Pb* for the bad state (see Fig. 3.3(b) and 3.3(c)). The bad state is characterized by a high probability of error, as opposed to the good state, whose probability of error is low. At each bit interval, the channel model changes states according to the transition probabilities: q_1 is the transition probability from the good state to the bad state while q_2 is the transition probability from the bad state to the good state. The errors occur in bursts with relatively long error-free intervals. The state transition are summarized by its transition probability matrix:

$$\begin{bmatrix} 1-q_1 & q_1 \\ q_2 & 1-q_2 \end{bmatrix} \tag{3.3}$$

The time intervals for the states *G* and *B* are both geometrically distributed with respective means $(q_2)^{-1}$ and $(q_1)^{-1}$ [40]. The steady state probability of being in the bad state is given by:

$$S_{pb} = S_{pg}(\frac{q_1}{q_2}) \tag{3.4}$$

where S_{pg} is the steady state probability of being in the good state.

For a two-state Markov model, the sum of all steady probabilities is equal to one:

$$S_{pb} + S_{pg} = 1 \tag{3.5}$$

The steady probability of being in the good state can be obtained from Eq. (3.5) as:

$$S_{pg} = 1 - S_{pb} = \frac{q_2}{q_2 + q_1} \tag{3.6}$$

The steady probability of being in the bad state can then be expressed as:

$$S_{pb} = \frac{q_1}{q_2 + q_1} = 1 - S_{pg} \tag{3.7}$$

The average BER is defined as the sum of all the products between the steady probability and the probability of error for each state and is given by:

$$BER = S_{pg} P_g + S_{pb} P_b \tag{3.8}$$

Substituting Eqs. (3.6) and (3.7) into Eq. (3.8) gives the average BER for a Gilbert-Elliot channel:

$$\overline{BER} = \left(\frac{q_2}{q_2 + q_1}\right) P_g + \left(\frac{q_1}{q_2 + q_1}\right) P_b = \frac{q_2 P_g + q_1 P_b}{q_2 + q_1} \tag{3.9}$$

3.1.2 The Rayleigh Fading Channel

A more sophisticated model for wireless channels should also take into account the possibility of a mobile receiver. When dealing with wireless communications and mobile receivers, the transmitted signal undergoes fades as the receiver moves. These types of channels are called fading multipath channels, since they present randomly time-variant impulse responses.

The first characteristic of a multipath medium is the time spread introduced in the signal that is transmitted through the channel. A second important characteristic is the time variations in the structure of the medium. These time variations appear to be unpredictable to the user of the channel and make the nature of the multipath vary with time.

For an arbitrary transmitted signal over a fading multipath channel, there exist multiple propagation paths. Associated with each path are a propagation delay and an attenuation factor. Both the propagation delays and the attenuation factors are time-variant as a result of changes in the structure of the medium. The multipath propagation model of the channel results in signal fading or amplitude variations in the received signal. If there are no fixed scatterers or signal reflectors in the medium, in addition to randomly moving scatteres, the channel is said to be a Rayleigh fading channel and the envelope of the received signal has a Rayleigh distribution. Fig. 3.4 shows the envelope of a received signal after transmission over a frequency non-selective, slow fading channel with a mobile speed of 3.6 *Km/ h* and a carrier frequency of 900 *MHz* .

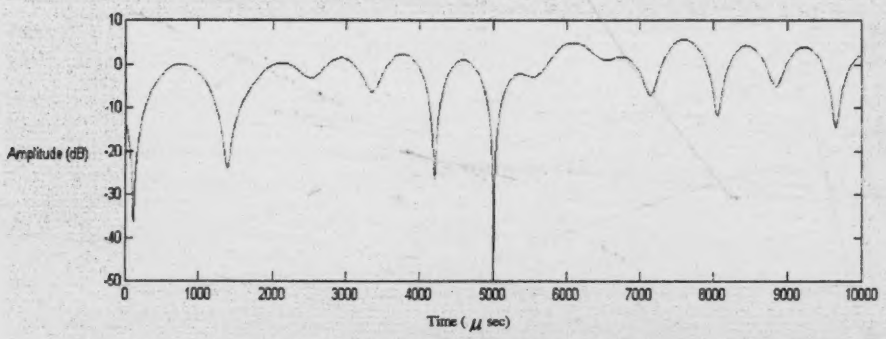


Figure 3.4. Envelope of a received signal after transmission over a frequency non-selective, slow fading channel with a mobile speed of 3.6 *Km/ h* and a carrier frequency of 900 *MHz* .

Two important concepts associated with a Rayleigh fading channel are the multipath spread and the Doppler spread of the channel. The multipath spread of the channel is defined as the range of values of time delay τ over which the average power output of the channel is essentially nonzero and is denoted by T_m. The Doppler spread of the channel is defined as the range of values of the Doppler frequency λ over which the signal intensity is essentially nonzero and is denoted by B_d.

The coherence bandwidth of the channel, defined as the frequency range across which fading properties are correlated and denoted by $(\Delta f)_c$ can be approximated to the reciprocal of the multipath spread [41] as:

$$(\Delta f)_c \approx \frac{1}{T_m} \tag{3.10}$$

Consequently, the channel affects two sinusoids with frequency separation greater than $(\Delta f)_c$ differently. A channel is said to be frequency selective if $(\Delta f)_c$ is smaller in comparison to the bandwidth of the information-bearing transmitted signal. If $(\Delta f)_c$ is larger than the bandwidth of the transmitted signal, the channel is said to be frequency-nonselective.

Similarly, the reciprocal of B_d is a measure of the coherence time of the channel, defined as the rate at which the channel characteristics changes and denoted by $(\Delta t)_c$ [41]:

$$(\Delta t)_c \approx \frac{1}{B_d} \tag{3.11}$$

A slowly changing channel has a large coherence time, or a small Doppler spread and it is said to be a slow-fading channel; that is, the channel characteristics vary sufficiently slowly that they can be measured.

Jakes [42] proposed a model to simulate Rayleigh frequency non-selective, slowly fading channels. In Jakes' model the fading envelope, *y(t)*, is realized by generating two filtered noise components, $X_c(t)$ and $X_s(t)$:

$$y(t) = X_c(t)\cos\omega_c t + X_s \sin\omega_c t \tag{3.12}$$

where $|y|$ is Rayleigh distributed with a probability density defined as $p(x) = (x/b)\cdot e^{-x/2b}$ for $x \geq 0$ and $p(x) = 0$ for $x < 0$, where $b = E_0^2/2$ is the mean power.

The two filtered noise components are comprised of a set of N sinusoidal signals as follows:

$$X_c(t) = 2\sum_{n=1}^{N_0} \cos\beta_n \cos\omega_n t + \sqrt{2}\cos\alpha\cos\omega_m t \tag{3.13}$$

$$X_s(t) = 2\sum_{n=1}^{N_0} \sin\beta_n \cos\omega_n t + \sqrt{2}\sin\alpha\cos\omega_m t \tag{3.14}$$

$$\omega_m = 2\pi v/\lambda \qquad \omega_n = \omega_m \cos\frac{2\pi n}{N} \qquad N_0 = \frac{1}{2}\left(\frac{N}{2} - 1\right) \qquad \theta = \frac{\pi}{4} \qquad \beta_n = \frac{\pi n}{N}$$

where ω_n are the Doppler shifts, ω_m is the maximum Doppler shift and N_0 indicates the number of low-frequency oscillators used in the simulations with $N \geq 6$.

The N_0 low-frequency oscillators with frequencies equal to the Doppler shifts plus one with frequency ω_m are used to generate signals that are frequency shifted from the carrier frequency ω_c. The weights applied to the frequency-shifted signals are chosen so to approximate the fading characteristics.

This work assumes frequency shift keying (FSK) as the digital modulation technique. However, the results can be extended to other modulation schemes such as binary phase shift keying (BPSK). Assuming FSK transmission over a channel with a fixed attenuation factor, the probability of error can be expressed as [41]:

$$P = Q\left(\sqrt{\alpha^2 \frac{E}{\eta}}\right) \tag{3.15}$$

where E is the energy, $\eta/2$ is the power spectral density, α is the fixed attenuation factor of the channel and $Q(x)$ is the Q function of x. Note that the attenuation factor accounts for the several causes of signal corruption in a wireless channel: signal attenuation due to distance, penetration losses through walls and floors, and multipath propagation.

This can be expressed in terms of the received signal-to-noise ratio (γ_R) per information bit as:

$$P = Q(\sqrt{\gamma_R}) = \frac{1}{2} erfc(\sqrt{\gamma_R/2}) \tag{3.16}$$

From [41], the error probability of the channel, p_e, when α is random can be expressed in terms of the average signal-to-noise ratio ($\bar{\gamma}$) as:

$$p_e = \frac{1}{2}\left(1 - \sqrt{\frac{\bar{\gamma}}{2+\bar{\gamma}}}\right) \tag{3.17}$$

As shown in Fig. 3.4, the signal envelope experiences very deep fades only occasionally. It is important to notice that a fade is more likely to occur as it becomes shallower. The level crossing rate, N_R, is used to describe this property and it is defined as the expected rate at which the envelope crosses a specified signal level, R, in the positive direction.

$$N_R = \sqrt{2\pi} f_{max\_Doppler} \rho e^{-\rho^2} \tag{3.18}$$

where $f_{max\_Doppler}$ is the maximum Doppler spread and is expressed as:

$$f_{max\_Doppler} = f_c \frac{v}{c} \tag{3.19}$$

with f_c as the carrier frequency, v the speed of the mobile, c the speed of light and ρ the received amplitude normalized by the RMS amplitude, expressed as:

$$\rho = \frac{R}{R_{RMS}} = \sqrt{\frac{SNR_R}{\overline{SNR}}} \tag{3.20}$$

where $\overline{SNR}$ is the average signal-to-noise ratio of the channel (dB) and SNR_R is the received signal-to-noise ratio (dB). Equation (3.20) shows that every time the fade envelope drops below $r=R$, a fade occurs. Let τ_i be the duration of the *ith* fade. Then, the average duration of fades for a total time interval of length T is

$$\bar{\tau} = \frac{1}{TN_R}\sum \tau_i = \frac{e\rho^2 - 1}{\rho f_D \sqrt{2\pi}} \tag{3.21}$$

where f_D is the normalized Doppler spread and is calculated by dividing the maximum Doppler spread by the data rate f_s.

Jakes' model provides the gain of the fading envelope according to the normalized Doppler spread and the *SNR* of the channel. The model can be represented as a black box with a set of input parameters and a set of outputs. This type of representation of a channel is called an input-output channel model. Fig. 3.5 depicts the realization of Jakes' model as an input-output channel model. Here, the $\overline{SNR}$ of the channel (in dB) and the normalized Doppler spread are inputs to the model. The output is the received envelope.

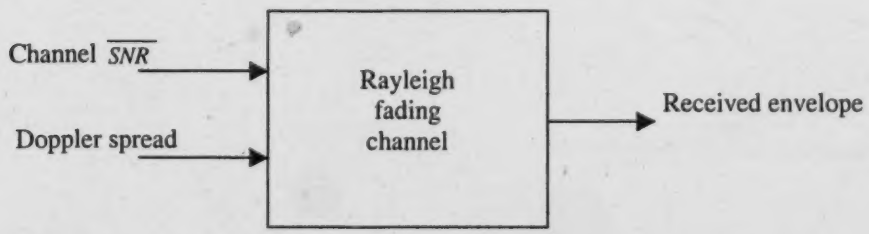


Figure 3.5. Realization of Jakes' model as an input-output channel model.

In order to simulate error patterns, the received signals with transmission errors are generated as:

$$r(t) = \alpha(t)s(t) \tag{3.22}$$

where $r(t)$ is the received signal before FSK demodulation, $\alpha(t)$ is the fading signal generated using Jake's model and $s(t)$ is the transmitted signal. The received bit-streams with errors are obtained after FSK modulation of the received signal. When a fade occurs, the demodulator cannot know if the signal information corresponds to a digital value of 1 or 0, and a burst of errors occur. Fig. 3.6-3.9 show the output of Jakes' model and the corresponding received bit-streams after FSK demodulation for four different channel conditions with a mobile speed of 3.6 Km/h, a carrier frequency of 900 MHz, a normalized Doppler spread of $6.7200e^{-4}$ and a SNR_R of 2.155 dB. A value of 1 in the bit-stream indicates a bit-error. Table 3.1 summarizes the theoretical and obtained level crossing rates and the average duration of the fades for the four channel conditions.

Table 3.1. Theoretical and obtained values for the fade durations of different fading channels. Normalized Doppler spread of $6.7200e^{-4}$ for FSK transmission

	$\overline{SNR}$	Theoretical Nr	Obtained Nr	Theoretical $\bar{\tau}$	Obtained $\bar{\tau}$
Simulation 1	10 dB	8/sec	8/sec	711 bits	515 bits
Simulation 2	11 dB	8/sec	9/sec	644 bits	643 bits
Simulation 3	15 dB	8/sec	7/sec	452 bits	399 bits
Simulation 4	20 dB	7/sec	7/sec	310 bits	236 bits

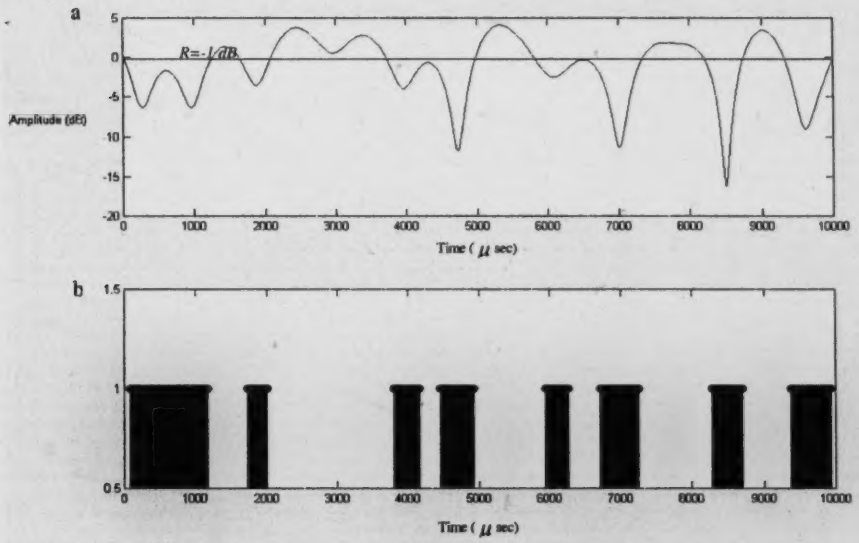


Figure 3.6. Simulation (1) of a Rayleigh fading channel using Jakes' model. a) Received fading envelope and b) burst of errors. $\overline{SNR}$ =10 dB, SNR_R = 2.155 (FSK), normalized Doppler spread = $6.7200e^{-4}$, signal level R= -1 dB.

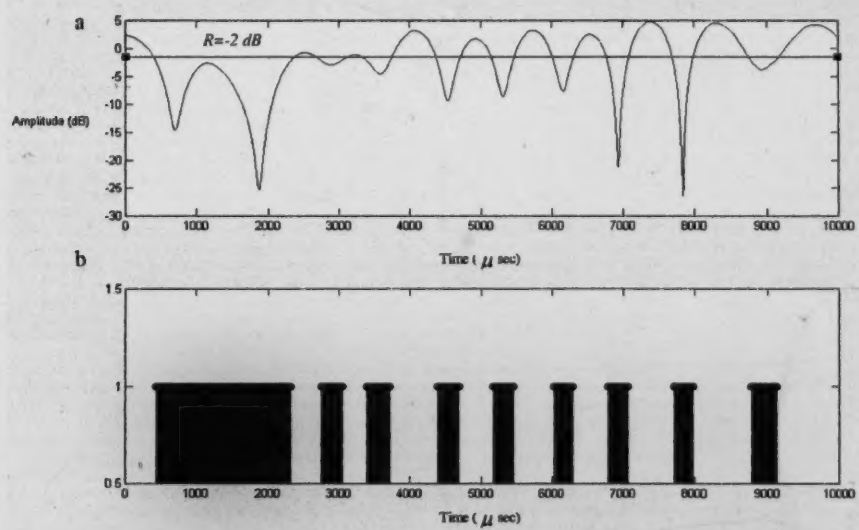


Figure 3.7. Simulation (2) of a Rayleigh fading channel using Jakes' model. a) Received fading envelope and b) burst of errors. $\overline{SNR}$ =11 dB, SNR_R = 2.155 (FSK), normalized Doppler spread = $6.7200e^{-4}$, signal level R= -2 dB.

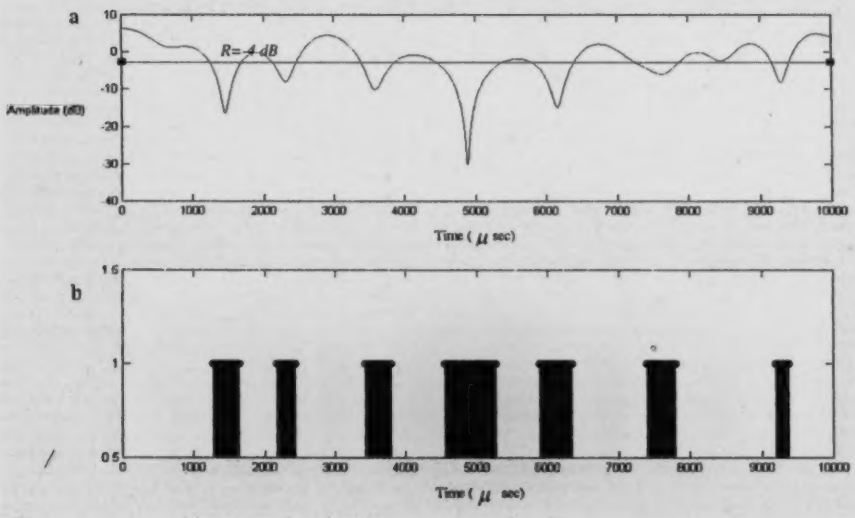


Figure 3.8. Simulation (3) of a Rayleigh fading channel using Jakes' model. a) Received fading envelope and b) burst of errors. $\overline{SNR}$ =15 dB, SNR_R = 2.155 (FSK), normalized Doppler spread = $6.7200e^{-4}$, signal level R= -4 dB.

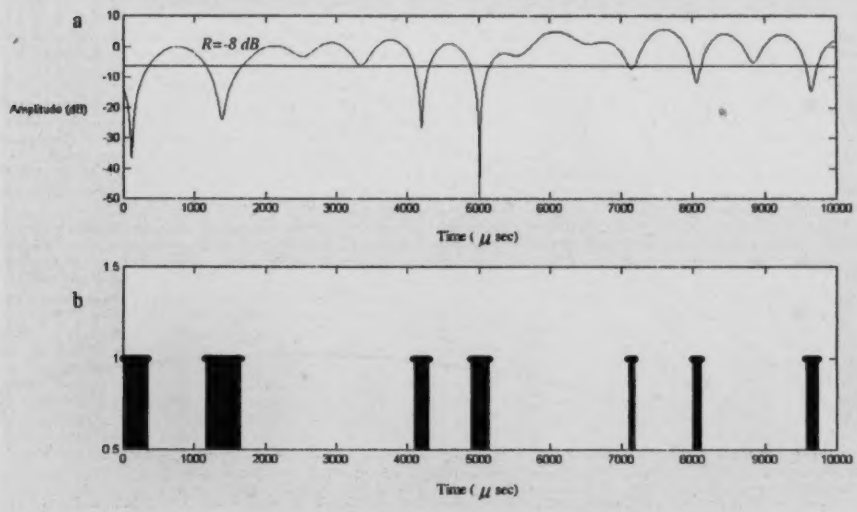


Figure 3.9. Simulation (4) of a Rayleigh fading channel using Jakes' model. a) Received fading envelope and b) burst of errors. $\overline{SNR}$ =20 dB, SNR_R = 2.155 (FSK), normalized Doppler spread = $6.7200e^{-4}$, signal level R= -8 dB.

The transmission environment used in this work is a personal mobile communication channel (transmission from a base station to a mobile receiver) simulated as a Rayleigh fading channel using Jakes' model. The received bit-streams with errors after demodulation are used to corrupt the compressed image data. FSK modulation with knowledge of the frequency and phase information of the original carrier signal is assumed in this work.

3.2 Channel Encoding Techniques

Channel encoding techniques are employed as a way to detect and correct errors in digital communications. The channel coding techniques can be classified into two major groups: block-codes and continuous-codes. Block codes divide the original message into blocks of fixed length. Redundant information is appended to each of these blocks. Examples of such block-codes are the cyclic redundancy check codes. On the other hand, continuous-codes append the redundant information with out dividing the original message. The encoding process is continuous and can be terminated at any time. Convolution codes are a common example of continuous-codes. In the following, the main characteristics of cyclic redundancy check codes and convolutional codes are described.

3.2.1 Cyclic Redundancy Check Codes

Cyclic Redundancy Check (CRC) codes are a subclass of Binary Cyclic Codes. They are widely used for error detection with long data packets and files in communication and computer applications. Cyclic codes have a well-defined algebraic structure and are easy to encode, which makes the decoding process easy. CRC codes work with blocks of data and append some redundant

information to the original message in order to detect and correct errors. In the following, the algorithm to encode a binary message using a CRC code is described.

The algorithm treats all bit streams as binary polynomials. For a given k-bit original message, the algorithm generates $m=n-k$ parity bits. The parity bits are appended to the original message. The resulting n-bit frame is the encoded message. The $m=n-k$ parity bits are generated so that the resulting encoded message is exactly devisable by some pre-defined polynomial. This pre-defined polynomial is called the devisor or CRC Polynomial. CRC codes are said to be cyclic because two important properties hold [38]:

i) The bit-by-bit addition of two codewords is again another codeword (linearity).

ii) Any cyclic shift of a codeword is also a codeword.

The first property means that CRC codes are linear and can be described as parity check codes. Due to this linearity property, a CRC code can be described as well as the set of all n-codewords generated by the matrix equation $c = iG$, where c is the codeword, i is any k binary vector and G is a k x n binary matrix called the generator matrix. Therefore, to encode a message using a CRC (n,k) code, it is necessary to either know the generator matrix and perform the operation $c = iG$ or perform the mathematical polynomial operations to obtain the parity bits to be appended to the original message. It is important to notice that CRC polynomial provides enough information to construct the generator matrix G.

Before describing the error detection and correction capabilities of CRC codes, it is important to define three important concepts: the Hamming distance, the Hamming weight and the minimum distance d of a linear code block. The

Hamming distance between two vectors having the same number of elements is defined as the number of positions in which the elements differ. On the other hand, the Hamming weight of a vector is the number of nonzero elements in the vector. The minimum distance d of a linear code bock is defined as the minimum Hamming weight of the nonzero codewords.

In order to detect and correct errors in the CRC-encoded message, it is important to find all the possible syndromes of the received message and generate a truth table for correction. A syndrome is found from the received codeword according to:

$$S = cH \tag{3.23}$$

where c is the received codeword and H is the m x n parity-check matrix defined as:

$$H = [G^T | 1] \tag{3.24}$$

The truth table contains all possible error pattern polynomials the CRC code is capable to correct. For a received codeword with t errors, an error pattern polynomial is a binary polynomial with t nonzero coefficients located where the errors have occurred. Each row in the truth table contains an error pattern polynomial. The truth table of a CRC code can be created using the following algorithm:

1. Generate all possible error pattern polynomials according to maximum number of errors, t, the code can correct. Let us remember that t can be obtained from the minimum distance d of the code according to $d = 2t + 1$

2. Corrupt a known original message with each of the possible error pattern polynomials. Generally, the original message is the all-zeros message.
3. Calculate all possible syndromes according to Eq. (3.23). Note that the result of Eq. (3.23) is a number in binary format.
4. Place each error pattern polynomial in the truth table in the row pointed by the corresponding syndrome. The first row is always row number zero and corresponds to an error free received codeword.

In order to correct a corrupted received codeword, it is necessary to first obtain the syndrome (the syndrome points to an error pattern polynomial in the truth table) and perform a modulo-2 addition between the corrupted codeword and the corresponding error pattern polynomial. The corrected message is obtained by removing the parity bits from the codeword.

3.1.2 Convolutional Codes

Block codes, such as CRC codes, use an encoding rule to associate $m = n - k$ parity symbols to each k-bit message. It is important to notice that the parity symbols, m, are created independently of others and depend only on those bits found in the same codeword. On the other hand, convolutional codes can encode a message without dividing it into blocks. Each symbol in the original message can affect a finite number of consecutive symbols in the output stream. Therefore, the resulting encoded stream is the result of a continuous encoding process that can be stopped at any time.

A convolutional code can be described by its rate and generator matrix. The rate of a convolutional code indicates the number of channel symbols to be associated to the message symbols. A rate k/n encoder associates n channel symbols to each

k information symbols. The generator matrix describes the way the channel symbols are generated and associated to the original message. Let us take a generator matrix $T = [5\ \ 7]$, in octal notation, for a rate-½ code. The generator matrix T can be expressed as a binary matrix as follows:

$$T = \begin{bmatrix} 101 \\ 111 \end{bmatrix} \tag{3.25}$$

The resulting encoded bit-stream is the multiplexation of different streams generated after the convolution of the original message with every element of the generator matrix T. To illustrate this, let us encode the message $m=[1\ 0\ 1\ 1\ 1]$ with a rate-½ code described by the generator matrix in Eq. (3.25). The encoded bit stream, c, is the multiplexation of $c_1 = m \otimes [101]$ and $c_2 = m \otimes [111]$ expressed as $c = [1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ ]$. It is important to notice that the length of the encoded bit-stream is not k times the length of the original message, as expected for a rate ½ encoder. This comes as a consequence of the convolution process.

A convolutional coder can be realized with shift registers and modulo-2 adders. Fig. 3.10 shows the realization of a rate-½ coder described by the generator matrix in Eq. (3.25).

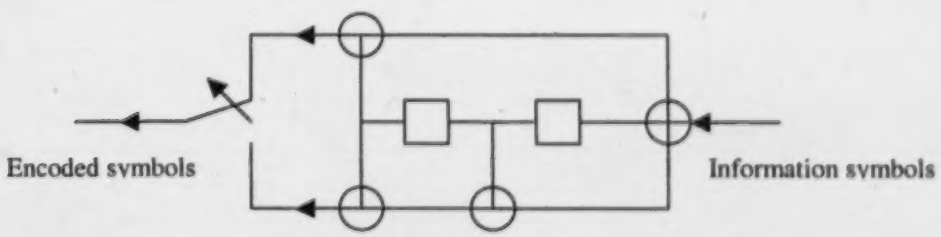


Figure 3.10. Rate-½ convolutional encoder.

The contents of the registers are assumed to be initially cleared, that is, they contain zeros. The information bits enter the coder from the right. In the left most side, the two encoded bit streams, corresponding to the two elements of the generator matrix, are multiplexed to create the final encoded bit stream. The diagram depicted in Fig. 3.10 has two shift registers and is said to be a memory-2 coder.

The simple convolutional coder depicted in Fig. 3.10 may be represented as well as a branching structure or code tree. By analyzing the encoder shown in Fig. 3.10, it is possible to obtain an encoding rule. The encoding rule describes the way the output bits are assigned according to the current input and previous encoded bits. Fig. 3.11 shows the code tree for the memory-2 coder shown in Fig. 3.10.

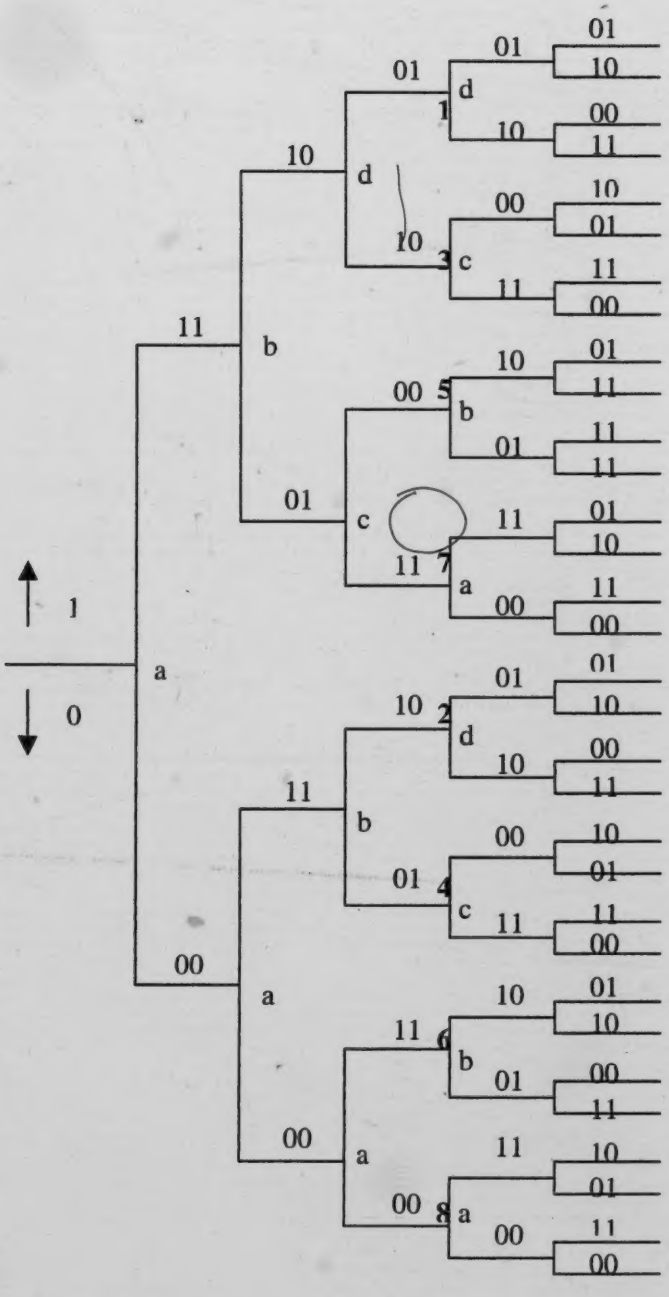


Figure 3.11. Code tree for a rate-½ convolutional coder.

In order to obtain the output channel bits, one begins at the root of the tree. If the first bit to be encoded is a 1, the encoder outputs 11. If it is a 0 bit, the encoder outputs 00. For the second bit, there are again two options, 1 or 0. If a one is to be encoded, the upper branch is read. If a zero is to be encoded, the lower branch is reached. This process is repeated until the final bit of the original message is reached. Note that the coding process can be stopped at any time.

A more compact way to represent the code tree depicted in Fig. 3.11 can be realized by exploiting the repetitive structure of the tree. Note that nodes 1 and 2 at the fourth level of the tree are related. The structures growing to the right of these two nodes are identical. The same happens with nodes 4-3, 5-6 and 7-8. These pairs of nodes can be merged into a single one to create a more compact representation of the code tree. This new representation has the form of a trellis. Fig. 3.12 shows the code trellis diagram for the code tree shown in Fig. 3.11.

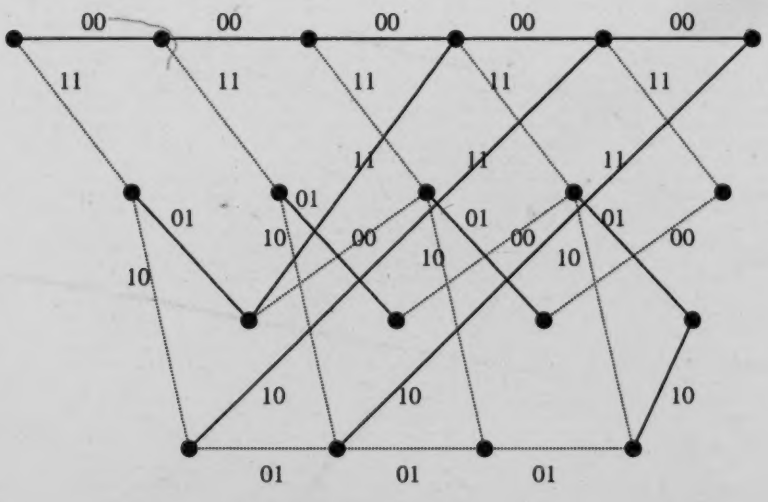


Figure 3.12. Code trellis diagram for the code tree shown in Fig. 3.11. A dashed line indicates the state transition due to an input bit of one.

Finally, there is another representation of a convolutional coder and code space useful to evaluate the performance of a code. This representation is in the form of a state diagram. A state diagram is a figure that shows the state transitions and the associated outputs for all possible inputs. In the case of a convolutional coder, the inputs are the different bits of the original message, the outputs are the corresponding codeword bits and the states are defined by the contents of the encoding shift registers (see Fig. 3.10). Fig. 3.13 shows the four-state diagram for the code tree shown in Fig. 3.11.

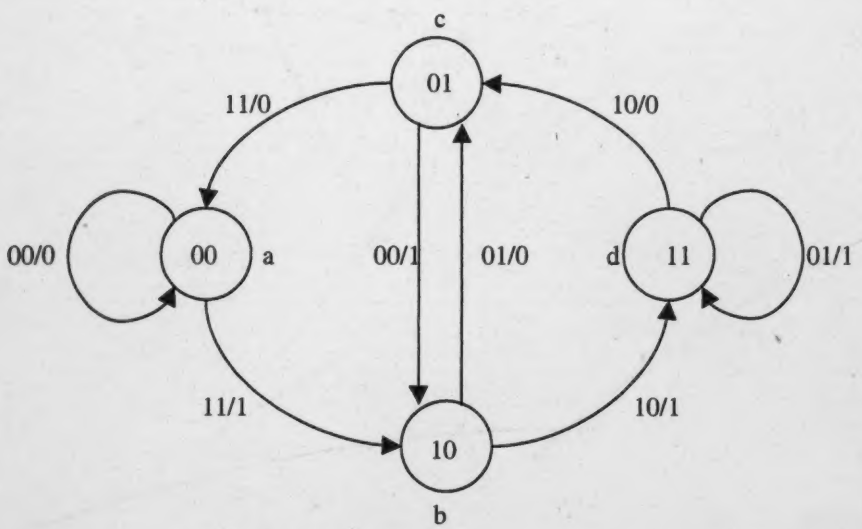


Figure 3.13. Four-state diagram for the code tree shown in Fig. 3.11.
State *a=00*, state *b=10*, state *c=01*, state *d=11*.

3.2.2.1 The Transfer Function of a Convolutional Code

The transfer function of a convolutional code can used to obtain some important distance properties of the code. These distance properties are useful to evaluate the performance of a code.

Let us take the state diagram depicted in Fig. 3.13 and assume that the all-zero sequence is the input to the encoder. First, the branches of the state diagram are labeled as D^0, D^1, or D^2, where the exponent of D denotes the Hamming distance of the output sequence of each branch from the output sequence of the all-zero branch. The state diagram is redrawn in Fig. 3.14 with the zero state, state *a* in this case, split into two states corresponding to the input, state *a*, and output, state *e*, of the state diagram.

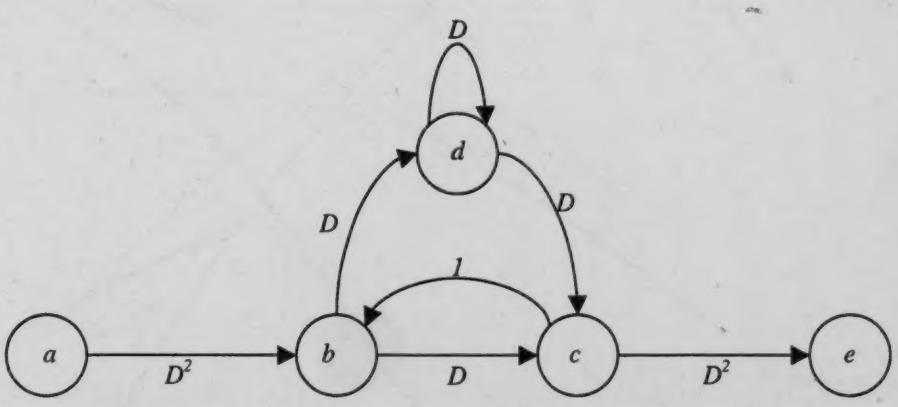


Figure 3.14. Modified state diagram for the convolutional coder shown in Fig. 3.13

This new diagram is used to obtain the four state equations:

$$X_b = D^2 X_a + X_c \tag{3.26}$$

$$X_c = DX_b + DX_d \tag{3.27}$$

$$X_d = DX_b + DX_d \tag{3.28}$$

$$X_e = D^2 X_c \tag{3.29}$$

The transfer function of a code is defined as $T(D) = X_e / X_a$. By solving the state equations given above, the transfer function for the code depicted in Fig. 3.14 is:

$$T(D) = \frac{D^5}{1-2D}$$

$$= D^5 + 2D^6 + 4D^7 + \ldots + 2^{d-5} D^d$$

$$= \sum_{d=5}^{\infty} a_d D^d \tag{3.29}$$

with $a_d = 2^{(d-5)}$.

The transfer function for this code indicates that there is a single path of Hamming distance $d=5$ from the all-zero path that merges with the all-zero path at a given node. The second term in the transfer function (see Eq. 3.29) indicates that there are two paths from node a to node e having a distance $d=6$. The third term indicates that there are four paths from node a to node e of distance $d=7$, and so forth. Hence, the transfer function provides the distance properties of a convolutional code. The minimum distance of a convolutional code is called the minimum free distance, denoted as d_{free}.

The transfer function can provide more detailed information about the code than just the distance of the different paths. Let us suppose two factors are introduced into the state diagram shown in Fig 23. A factor N placed in a branch only if a branch transition is due to an input bit 1. And a factor L placed in each branch so that its exponent serve as an accounting variable to indicate the number of branches in any given path from node a to node e. The new state diagram with the additional factors N and L is shown in Fig. 3.15.

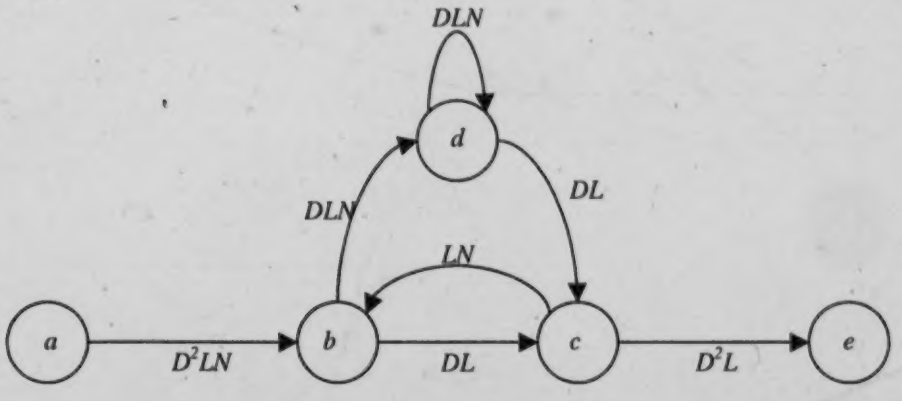


Figure 3.15. Modified state diagram for the convolutional coder shown in Fig. 3.14

The new state equations are then expressed as:

$$X_b = LND^2X_a + LNX_c \tag{3.30}$$

$$X_c = LDX_b + LDX_d \tag{3.31}$$

$$X_d = LNDX_b + LDX_d \tag{3.32}$$

$$X_e = LD^2X_c \tag{3.33}$$

and the transfer function is:

$$T(D,L,N) = \frac{D^5 L^3 N}{1 - DLN(1+L)}$$

$$= D^5 L^3 N + D^6 L^4 (1+L) N^2 + D^7 L^5 (1+L)^2 N^3 + \ldots$$
$$+ D^{k+5} L^{k+3} (1+L)^k N^{k+1} + \ldots \qquad (3.34)$$

This new representation of the transfer function gives the properties of all paths in the convolutional code. For example, the first term in Eq. (3.34) indicates that the distance $d=5$ path is of length 3 and it differs from the all-zero information sequence by one bit.

When an information sequence of finite duration, say n bits is encoded using a convolutional code, the factor L is important, as it indicates that the transfer function for the finite sequence is obtained by truncating $T(D,L,N)$ at the term L^n. However, if a extremely long sequence is encoded, it is advisable to suppress the dependence of $T(D,L,N)$ on the parameter L. This is usually done by setting $L=1$. For the transfer function in Eq. (3.34), one has:

$$T(D,1,N) = T(D,N)\frac{D^5 N}{1 - 2DN} \qquad (3.35)$$

3.2.3 Decoding Convolutional Codes

The main objective of a convolutional decoder is to find the codeword in the set of all possible transmitted codewords that most closely resembles the transmitted information. Maximum likelihood decoding is the most used method to decode

convolutional codes. It involves searching all the possible transmitted codewords and finding the one that is most closely in Hamming distance to the received data. Maximum likelihood is impractical when dealing with large encoded bit-streams. Let us suppose an original message with a 100-bit length is encoded with a rate-½ convolutional encoder. The set of all possible transmitted codewords contains 2^{100} different code-streams. Searching this set is time consuming and requires a large amount of resources in order to store the different code-streams. The Viterbi [43] algorithm provides a maximum likelihood decoding procedure that reduces the size of the set of possible transmitted codewords dramatically by exploiting the repetitive structure of the code tree. Following, the Viterbi algorithm is described for a rate-½ code with a generator matrix described in Eq. (3.25).

3.2.3.1 The Viterbi Decoding Algorithm

Let us assume the first six received bits of a code stream are 01 01 10 and consider the code tree shown in Fig. 3.11. The bits along the two paths through the code tree ending at points 1 and 2 are 11 10 01 and 00 11 10 respectively. The fist sequence of bits is at Hamming distance 5 from the received code-stream. The second sequence of bits is at Hamming distance 2 from the received code-stream. Both paths end at node *d*, and from this node, they see the same set of branches growing off to the right. Consequently, there is no way the first sequence can get closer to the second sequence, since the remaining bits will disagree and agree in the same way with the received information. Therefore, all the paths growing off node 1 can be deleted. The same analysis can be applied to paths growing off nodes 4 and 3, those growing off nodes 5 and 6 and those growing off nodes 7 and 8. The whole process can be repeated at the fifth and subsequent levels of the code tree. The paths not eliminated in the end are called survivors. The survivor with the shortest Hamming distance is considered as the correct codeword. The main idea behind the Viterbi algorithm is that the number of paths extended and

retained remains constant and equals the number of states of the coder. The correct codeword is in that set of retained paths.

The Viterbi decoding algorithm reduces the number of computations needed and works best when the errors found in the received code-stream are randomly distributed. This kind of error patterns is found in a binary symmetric channel. However, errors tend to occur in bursts in a fading channel. If burst of errors are encountered, the use of a convolutional interleaver before transmitting the symbols is advisable. A convolutional interleaver is a triangular array used to scramble the encoded information before transmission. Each element of the array is used to store one code symbol. The symbols are read into the interleaver before transmission. Fig. 3.16 depicts a convolutional interleaver of depth 7. A depth of a convolutional interleaver is defined as the number of rows of the triangular array. Inspection of the diagram suggests that adjacent bits in the encoded bit-stream are separated by at least 7 bits in the interleaved bit-stream. The use of a convolutional interleaver reduces the memory associated with a bursty channel making it appear as an independent-error channel. The penalty paid is the delayed introduced in the transmission, which is d^2 bits, where d is the depth of the interleaver.

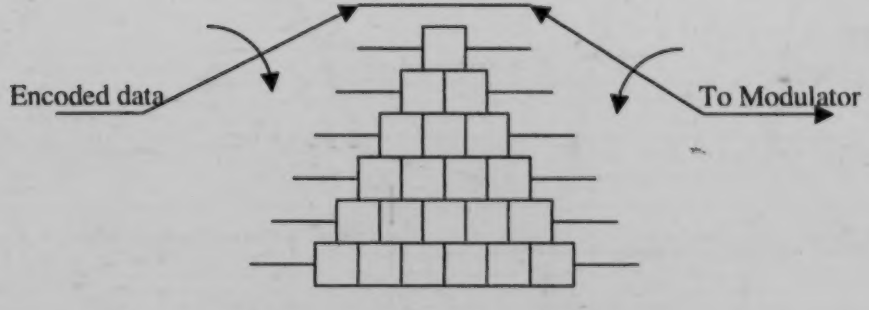


Figure 3.16. Depth-7 convolutional interleaver.

3.2.4 Rate Compatible Convolutional Codes

It has been described how convolutional encoders can encode arbitrary length messages resulting in long code sequences. The Viterbi algorithm has also been described as a method to decode convolutional code-streams. However, sometimes the transmission channel imposes bit-rate constrains. If a specific bit budget has to be met, it is preferable to reduce the rate of the convolutional encoder to obtain shorter code streams. This process can be achieved by using a Rate Compatible Punctured Convolutional (RCPC) code [44].

A RCPC is a convolutional code where the output code stream has been punctured according to a puncturing $n \times P$ matrix defined as:

$$a(l) = [a_{ij}(l)] \tag{3.36}$$

with $a_{ij}(l) \in (0,1)$, where 0 implies puncturing. P is said to be the puncturing period and $R=1/n$ is the rate of the code to be punctured or mother code. The puncturing period P, along with n, determines the range of code rates between $P/(P+1)$ and $1/n$ that can be obtained from the mother code as follows:

$$R = \frac{P}{P+l} \qquad l = 1,.....,(n-1)P \tag{3.37}$$

Let us take the rate-½ mother code with transfer function $T = \begin{bmatrix} 101 \\ 111 \end{bmatrix}$. Let us assume we have a binary puncturing matrix a with a puncturing period $P=4$, defined as:

$$a = \begin{bmatrix} 1001 \\ 1110 \end{bmatrix} \tag{3.38}$$

It has been shown that the resulting code-stream is the multiplexation of two streams, *c1* and *c2*. The first stream *c1* is the binary convolution between the original message and 1 0 1 (e.g. the first element of the transfer function) while the second stream *c2* is binary convolution between the original message and 1 1 1. The first element of matrix *a*, ([1001]), punctures stream *c1*. This first element indicates that every second and third bit of stream *c1* is not to be transmitted. The second element of matrix *a*, ([1110]), punctures stream *c2*. This second element indicates that every fourth bit of stream *c2* is not to be transmitted. By using these puncturing rule, a rate-*4/5* code has been achieved. If this code is not powerful enough to correct errors in the received code word, extra bits may be transmitted with no need to retransmit the whole code stream. The decoder knows the puncturing matrix and only uses the transmitted bits to compute the Hamming distances between the received code-stream and the possible correct messages.

3.2.5 Concatenation of CRC and RCPC Codes

A more powerful low-rate channel code can be realized by concatenating two or more relatively simple codes. Fig. 3.17 depicts an example of the concatenation of a CRC code and a RCPC code. The information to be transmitted is first encoded with a CRC *(n,k)* code, which is called the outer code. The coded symbols are then treated as the input to the RCPC code, which is called the inner code. At the receiver side, the demodulated data is first decoded with a decoder for the inner code, and the output of this is then decoded with a decoder for the outer code.

Some important characteristics of concatenated CRC and RCPC codes are [38]:

i) The resulting overall codeword length can be larger than the codeword length of the inner or outer code.

ii) The resulting code rate, consequently, can be quite low. This rate is the product of the individual code rates.

iii) The decoding process can be implemented in stages, reducing the complexity.

iv) The outer coder must be chosen to properly decode the bit-errors that remain after decoding the inner code. If the outer coder rate is too high, residual bit-errors cannot be reliable corrected.

The most suitable channel codes for use as inner codes are those that use maximum likelihood decoding, therefore short–constraint-length convolutional codes are good candidates. The constraint length, K, of a convolutional code is defined as $K=m+1$, with m as the memory of the code. Fig. 3.10 shows a constraint-3 rate ½ convolutional encoder.

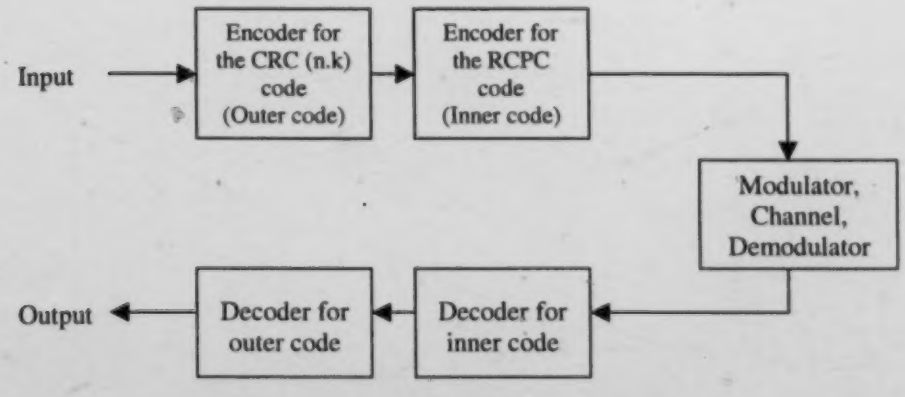

Input
Encoder for the CRC (n.k) code (Outer code)
Encoder for the RCPC code (Inner code)
Modulator, Channel, Demodulator
Output
Decoder for outer code
Decoder for inner code

Figure 3.17. Concatenation of a CRC code and a RCPC code.

3.2.6 Performance of RCPC Codes over Rayleigh Fading Channels

In order to optimally assign channel protection by means of a RCPC code to any binary information, it is important to know the performance of the convolutional code over the transmission environment. In the particular case of this work, the transmission environment is that of a personal mobile communication channel (Rayleigh fading channel) simulated with Jakes' model as described in section 3.1.3. In this section, the performance of the RCPC codes over a Rayleigh fading channel is reviewed. The analysis assumes the use of an interleaver on the encoded information prior transmission in order to erase the memory associated with the channel.

The fist step in determining the performance of a convolutional code over a noisy channel is finding the first-event error probability, which is defined as the probability that the correct path is excluded for the first time at the *jth* step, and the bit-error probability.

Let us assume that the all-zero sequence is transmitted over the channel and the probability of error in deciding in favor of another sequence is to be determined. Moreover, let suppose that the path being compared with the all-zero sequence at node X has Hamming distance d from the all-zero path. If d is odd and the number of errors in the received sequence is less than $(d+1)/2$, the all-zero path is selected; otherwise, an incorrect path is selected. Hence, the probability of selecting the incorrect path is:

$$P_d = \sum_{k=(d+1)/2}^{d} \binom{d}{k} p^k (1-p)^{d-k} \tag{3.39}$$

where p is the error probability of the channel expressed in terms of the average signal-to-noise ratio ($\overline{SNR}$). For FSK transmission, p is given in Eq. (3.17).

If d is even, an incorrect path is selected when the number of errors exceeds $d/2$. Therefore, the probability of selecting an incorrect path over the all-zero path is:

$$P_d = \sum_{k=d/2+1}^{d} \binom{d}{k} p^k (1-p)^{d-k} + \frac{1}{2}\binom{d}{d/2} p^{d/2}(1-p)^{d/2} \tag{3.40}$$

where the second term in Eq. (3.40) indicates the probability of selecting a wrong path when the number of errors equals $d/2$. Since there are many paths with different distances that merge with the all-zero path at a given node and the first-event error probability is difficult to find, this error probability can be upper bounded [41] by:

$$P_e < \sum_{d=d_{free}}^{\infty} a_d P_d \tag{3.41}$$

where the coefficients $\{a_d\}$ are given in the expansion of the transfer function $T(D)$ and P_d is as defined in Eqs. (3.39-3.40). As for the probability of error, the transfer function $T(D,N)$ can be used to obtain the expression for the upper bound on the bit error probability, as:

$$P_b < \sum_{d=d_{free}}^{\infty} c_d P_d \tag{3.42}$$

where the coefficients $\{c_d\}$ are the coefficients in the expansion of the derivate of *T(D,N)* evaluated at *N=1*. Let us remember that the exponents in the factor of *N* indicate the number of nonzero bits that are in error when an incorrect path is selected over the all-zero path. By differentiating *T(D,N)* with respect to *N* and setting *N=1*, the exponents of *N* become multiplication factors of the corresponding error event probabilities P_d [41].

For a family of RCPC codes, there is a set of discrete channel coding rates, given by:

$$\xi = \{r_1, r_2, r_3, \ldots r_{(n-1)P}\} \tag{3.43}$$

where n is the rate of the mother code defined as $R=1/n$, and P is the puncturing period.

Each of the channel coding rates in vector ξ has an associated bit error probability (see Eq. 3.42) according to the channel conditions. The set of error probabilities is given as:

$$\delta = \{p_b^1, p_b^2, p_b^3, \ldots p_b^{(n-1)P}\} \tag{3.44}$$

where p_b^n is the upper bounded bit error probability for the discrete channel coding rate n in vector ξ.

The upper bounded bit error probabilities in vector δ are used to design the channel protection techniques proposed in this work.

3.3 Summary

The main characteristics of CRC codes and RCPC codes were presented in this chapter, emphasizing the error detection and correction capabilities of the codes. First, some common channel models used to simulate different transmission environments were described, including the binary symmetric channel and Gilbert-Elliot channel model. Jakes' model was presented as a way to simulate Rayleigh fading channels, commonly found in personal mobile communication systems. The realization of Jakes' model as an input-output channel model was introduced and some simulations were performed to evaluate the performance of the model under different channel conditions. Following, the performance analysis of convolutional codes over ideally interleaved channels was described. This analysis showed that for a family of RCPC codes, it is possible to obtain different probabilities of error according to the channel conditions.

Chapter 4

Efficient Channel Protection for JPEG2000 images

This chapter discusses adding controlled redundancy through FEC to the JPEG2000 images with no modification to the source coding process. The error resilient tools adopted in JPEG2000 and the hierarchical structure of the JPEG2000 bit-stream are first reviewed, followed by an analysis of the effect of channel errors in code-blocks and packets of the JPEG2000 bit-stream. This analysis is used as the basis to derive different channel protection techniques optimized for the channel conditions. The first technique presented is an equal channel protection technique. The technique assumes a layered JPEG2000 bit-stream with one tile, one component, a PPM marker segment and a progression by resolution order. The channel protection is optimally assigned to all the data packets in the bit-stream with knowledge of the channel conditions. The transmission environment used in this work is a personal mobile channel simulated as Rayleigh fading channel using Jakes' model as described in section 3.1.3.

4.1 Error-resilient Tools in JPEG2000

The JPEG2000 standard [32] has made a considerable progress to address the issue of communications over error prone channels. As described in section 2.6, the JPEG2000 standard makes use of a context based arithmetic coder to compress the coefficient bit-planes of the different code-blocks. Although the arithmetic coder provides a good compression performance, the resulting coded data is highly prone to bit-errors [1]. A single bit-error can produce the loss of synchronization at the decoder side resulting in error propagation and corrupted decoded information. A set of error resilient tools has been included in the JPEG2000 standard to reduce the impact of transmission errors on compressed images. The tools work at the entropy coding and packet level, and have been shown to provide a better performance compared to the case of the use of no error resilient tools [45]. The error resilience tools deal with channel errors using the following approaches: data partitioning and resynchronization, error detection and concealment, and Quality of Service (QoS) transmission based on priority.

4.1.1 Data Partitioning and Resynchronization

Resynchronization tools are designed to establish the synchronization between the decoder and the bit-stream. A good resynchronization tool should be able to localize the error and prevent it from affecting the entire bit-stream. Inserting markers along the bit-stream is a common way to achieve this. Once the synchronization has been established, the data between the synchronization point prior the error and the point where synchronization is reestablish is usually discarded. Fig. 4.1 shows the way synchronization is reestablished after a bit error is found in an entropy-coded bit-stream.

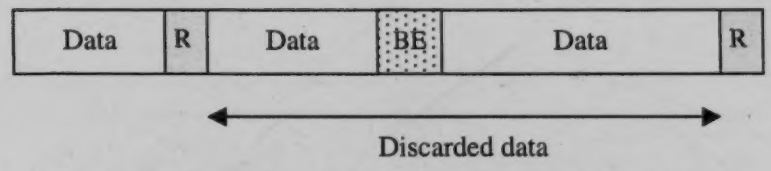


Figure 4.1. Resynchronization after a bit-error is found in an entropy-coded bit-stream. R: Resynchronization marker. BE: bit-error.

A data partitioning strategy can be used as an approach to resynchronization. This strategy divides or arranges the bit-stream into sections according to their sensitivity to errors. Data partitioning is based in the assumption that the same bit-error in two different sections of the bit-stream can have a different impact. In the JPEG2000, the encoded information is organized in a hierarchical way. This hierarchical structure will be described in section 4.2.

4.1.2 Error Detection and Concealment

Error detection and concealment strategies rely on the resynchronization tools and its ability to localize errors. Error concealment methods attempt to improve the quality of the reconstructed image using the information obtained from the error detection and resynchronization tools [1] and the image data itself. The JPEG2000 standard does not specify any error concealment methods and its design and usage are decoder dependent.

Table 4.1 summarizes the tools for error-resilience [32] in the JPEG2000 standard.

Table 4.1. Error resilient tools in JPEG2000

Type of tool	Name
Entropy coding level	1. Independent coding of code-blocks 2. Termination and reset of the arithmetic coder for each pass 3. Selective arithmetic coding bypass (lazy coding mode) 4. Segmentation symbols
Packet level	5. Short packet format 6. Resynchronization marker

The error resilient tools work as follows:

1. Quantized coefficients of different code-blocks are entropy coded independently and errors in the coded data of a code-block are contained within that code-block.
2. The arithmetic coder can be terminated and reset after each coding pass. This allows the arithmetic coder to continue to decode coding passes even in the presence of errors.
3. The optional arithmetic coding bypass style puts raw bits into the code-stream without arithmetic coding. This style of coding allows bypassing the arithmetic coder for the significance propagation pass and magnitude refinement coding passes in the fifth significant bit-plane and the following bit-planes, of a code-block. This prevents error propagation to which entropy coded data is highly susceptible.
4. A segmentation symbol is a special symbol that is coded at the end of each bit-plane. The correct decoding of this symbol confirms the correct reception of a

bit-plane.

5. At the packet level, short packets are achieved by moving the packet headers to the main header in the PPM or PPT marker segments. In the case of errors in the packets, the packet headers in the PPM or PPT marker segments can still be associated with the correct packet by using the sequence number in the SOP.
6. A SOP resynchronization marker is inserted in front of every packet with a sequence number starting at zero. This number is incremented with each packet and it is useful to identify missing packets in packet-erasure prone networks.

Although the error resiliency tools can improve the resiliency to errors of the JPEG2000 bit-stream, their usage does not guarantee an error-free received image, since residual bit-errors can still affect the compressed information. Therefore, channel protection is still advisable when dealing with error prone channels. One way to reduce or eliminate the effect of error prone channels is the use of forward error correction and automatic repeat request [1]. However, the joint use of a FEC-ARQ technique involves the introduction of additional delay, which is not always desirable in practical applications. The use of a stand-alone FEC technique can efficiently protect the coded information against transmission errors, without the introduction of additional delays.

4.2 Hierarchical Structure of the JPEG2000 Bit-stream

The general structure of a JPEG2000 bit-stream for one tile with one component and n packets is depicted in Fig. 2.10 in section 2.8. This figure is repeated here for convenience.

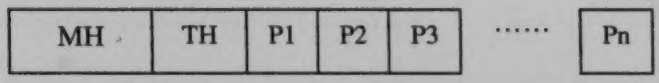


Figure 4.2. Organization of a JPEG2000 bit-stream for one tile and *n* packets. MH: Main Header. TH: Tile-part Header. Pn: *nth* Packet.

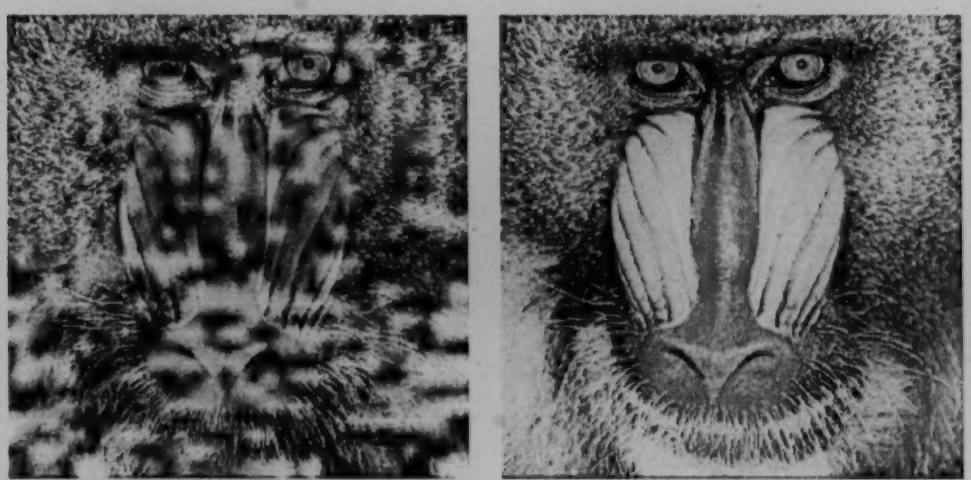
The main header of the JPEG2000 bit-stream contains all the necessary information needed to correctly decode the image data. This information is organized in markers segments, as described in section 2.8. Therefore, the main header is the most important part of the final bit-stream. A transmission error in the main header can result in the failure of the whole decoding process.

The packets that conform the layers used to encode the image data comprise the rest of the bit-stream. If single-quality-layer compression is used, the initial packets contain the compressed information corresponding to the most significant coefficients (e.g. the coefficients from the low frequency sub-bands), while the later packets carry the details of the image (e.g. the coefficients from the high frequency sub-bands). In natural images, most of the energy of the wavelet transform is concentrated in the LL_L sub-band [46]. Moreover, the energy in the high frequency sub-bands is concentrated in a small number of coefficients. Hence, the information carried in the initial packets of the JPEG2000 bit-stream tends to increase the distortion of the reconstructed images if transmission errors occur. A common quality measure for image communication applications is the peak signal-to-noise ratio (PSNR), which for gray-level images is defined as:

$$PSNR = 10\log_{10} \sum_{i,j} \frac{255^2}{\frac{1}{N}(p(i,j) - \tilde{p}(i,j))^2} \tag{4.1}$$

where $p(i, j)$ are the pixel values in the original image, $\tilde{p}(i, j)$ are the pixel values in the test image and N is the total number of pixels in the image.

Fig. 4.3 shows the effect of bit-errors in the LL_L and *HH1* sub-band with $L=4$ in a JPEG2000 encoded image. Note the different impact in the quality of the reconstructed images.

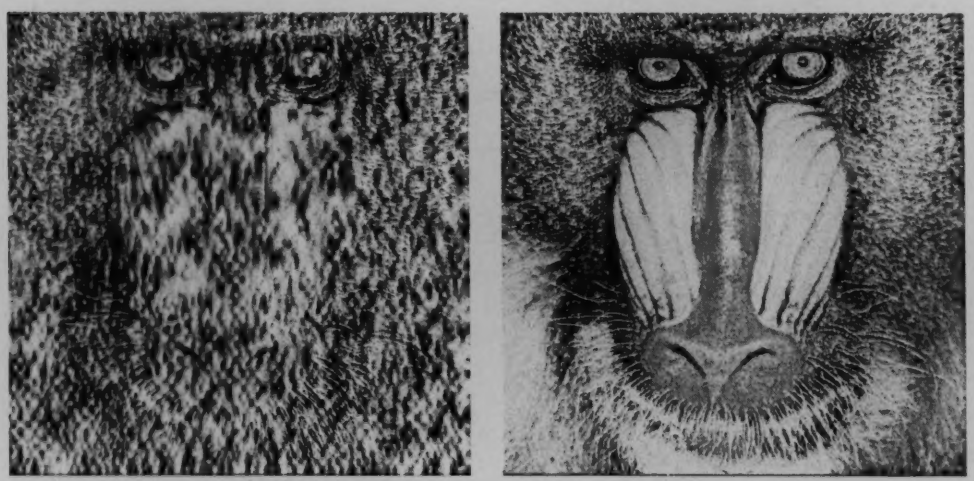


(a) *PSNR=13.2238 dB* (b) *PSNR= 63.8386 dB*

Figure 4.3. Effect of bit-errors in the (a) LL_L and (b) *HH1* sub-band with $L=4$ in the gray-level Baboon image encoded by means of the JPEG2000 standard with one quality layer.

When the image is compressed using multiple quality layers, the coded data of each code-block is distributed across a number of layers. As mentioned in section

2.7, each layer consists of a few numbers of consecutive bit-plane coding passes from each code-block, including all sub-bands [1]. The initial layers, and therefore the initial packets, contain the initial coding passes of each code-block. These initial coding passes correspond to the most significant bit-planes of the code-blocks. The later layers, and therefore the later packets, contain the coding passes corresponding to the least significant bit-planes of each code-block. Therefore, bit-errors in the initial packets tend to significantly increase the distortion of the reconstructed image. Fig. 4.4 shows the effect of bit-errors in the initial layer and final layer in a JPEG2000 encoded image with five quality layers. Note again the different impact in the quality of the reconstructed images.

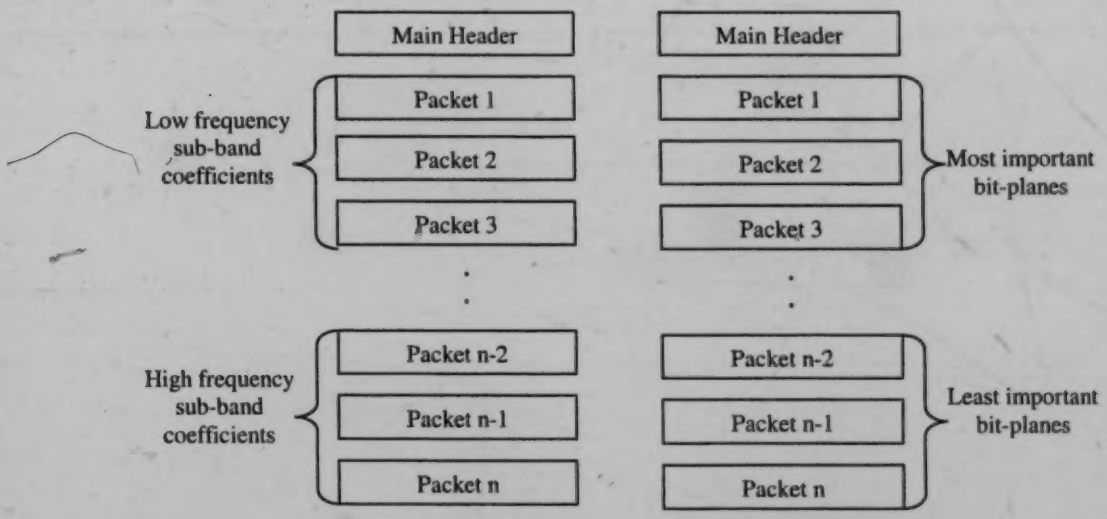


(a) *PSNR=15.0744 dB* (b) *PSNR=61.75172 dB*

Figure 4.4. Effect of bit-errors in the (a) initial layer and (b) final layer in the gray-level Baboon image encoded by means of the JPEG2000 standard with five quality layers.

The hierarchical structure of the JPEG2000 bit-stream for single and multiple-quality-layer compression is depicted in Fig. 4.5. It has been assumed one tile with

one component and n packets. The tile-part header has been included in the main header for simplicity.



a) Single-quality-layer compression b) Multiple-quality-layer-compression

Figure 4.5. Hierarchical structure of the JPEG2000 bit-stream for a) single and b) multiple-quality-layer compression.

It has been demonstrated in [33] that multiple-quality-layer compression performs better under noisy channel conditions and FEC techniques. Therefore, this work assumes a multiple-quality-layer compression scheme.

4.3 Effect of Bit-errors on the JPEG2000 Bit-stream

In this section, an analysis of the effect of bit-errors in code-blocks and packets of the JPEG2000 bit-stream is presented. Expressions for the effect of channel errors in code-blocks and packets, in terms of the mean square error (MSE), are derived.

In this work, the distortion of the reconstructed image has been estimated from the MSE in the wavelet domain. The estimation is exact for orthogonal wavelets. Although the default JPEG2000 wavelets are generally bi-orthogonal [32] (for which the Parseval's power theorem is not valid) in nature, we still assume orthogonality condition. For near-orthogonal wavelets, the analysis is expected to provide a very close performance to the orthogonal case.

4.3.1 Effect of Bit-errors in a Code-block

Because of the nature of the coding passes and the arithmetic entropy coding process, the magnitude of the distortion of the reconstructed image depends on the number and position of the bit-errors. A bit-error in the initial few bits of a code-block coded data generally results in a higher distortion compared to a bit-error in the later bits. This is illustrated in Fig. 4.6. Here, the gray-level 512×512 Lena image was compressed by means of the JPEG2000 standard with four levels of decomposition using the default reversible wavelet transform and a single quality layer. No error resilient tools were employed. The code-block size was set to 64×64 samples and the precinct to 512×512 samples. A single bit-error was simulated in the fist byte of the first code-block code data of the bit-stream (this code-block corresponds to the LL_4 frequency sub-band). The reconstructed image is shown in Fig. 4.6(a). Fig. 4.6(b) shows the reconstructed image after a single bit-error in the last byte of the first code-block coded data of the bit-stream. Notice the quality difference, in terms of the PSNR, between the two reconstructed images. This quality difference shows that the worst case is the first bit-error occurring in the initial bits of the coded data of a code-block. Errors after this first bit-error will gradually increase the value of the MSE for that code-block, until the maximum MSE value is reached (e.g. errors in the different bit-planes of the code-block).

The MSE, for a code block b in sub-band s (hereafter referred to as code-block (b, s)) can be expressed as:

$$m_{b,s} = \frac{(\Delta_{b,s})^2}{N_{b,s}} \sum_{n=1}^{N_{b,s}} (C_n - \tilde{C}_n)^2 \tag{4.2}$$

where C_n is the nth quantized coefficient in code-block (b,s), $\tilde{C}_n$ is the corresponding corrupted coefficient, $N_{b,s}$ is the total number of samples in code-block (b, s) and $\Delta_{b,s}$ is the quantization step for code-block (b, s).

(a) *PSNR=8.9768 dB* (b) *PSNR= 68.9536 dB*

Figure 4.6. Effect of a single bit-error in the initial code-block coded data of a JPEG2000 bit-stream. a) The single bit-error has been simulated in the first byte of the initial code-block coded data. b) The single bit-error has been simulated in the last byte of the initial code-block coded data.

If there is an error in every bit-plane and the sign of every coefficient of code-block (b, s), the MSE in Eq. (4.2) is the maximum MSE (MMSE and denoted as $\mathrm{M}_{b,s}$ hereafter). The MMSE for code-block (b, s) on a per pixel basis (i.e. over the entire image and denoted as $\overline{\mathrm{M}}_{b,s}$) can be calculated as:

$$\overline{\mathrm{M}}_{b,s} = \frac{s_{b,s}}{B_s \cdot S} \cdot \mathrm{M}_{b,s} = \frac{2^{-2r_{b,s}}}{B_s} \cdot \mathrm{M}_{b,s} \tag{4.3}$$

where S is the total number of image pixels, $r_{b,s}$ is the resolution of sub-band s (r=1 corresponds to the first level of decomposition), $s_{b,s} = S/2^{2r_{b,s}}$ is the number of coefficients in sub-band s and B_s is the number of code-blocks in the sub-band (the code-blocks are of equal size).

Assuming an error in every bit-plane and the sign of every coefficient, the $\overline{\mathrm{M}}_{b,s}$ with B_s code-blocks of equal size, can be expressed as follows:

$$\overline{\mathrm{M}}_{b,s} = \frac{2^{-2r_{b,s}} \cdot \mathrm{M}_{b,s} \cdot \Delta^2}{B_s} \tag{4.4}$$

where

$$\mathrm{M}_{b,s} = \frac{\sum_{n=1}^{N} \left[\varphi(C_{n,b}) \cdot \sum_{j=0}^{p-1} \phi_j(C_{n,b}) \cdot 2^{2j} \right]}{N_{b,s}}$$

$$\varphi(C_{n,b}) = \begin{cases} 1 & if \quad C_{n,b} \le 0 \\ -1 & if \quad C_{n,b} > 0 \end{cases}$$

$$\phi_j(C_{n,b}) = \begin{cases} 1 & \text{if the } jth \text{ bit of coefficient } C_{n,b} \text{ is 0} \\ & \text{where } j = 0 \text{ is the LSB and } j = p-1 \text{ is the MSB} \\ 0 & \text{otherwise} \end{cases}$$

Let assume a 3×3 code-block contains the following quantized coefficients:

14	3	5
1	-2	7
-4	12	8

Figure 4.7. A 3×3-sample code-block with nine quantized coefficients.

The bit-plane representation of the quantized coefficients is illustrated in Table 4.2.

Table 4.2. Bit-plane representation of quantized coefficients.

Coefficient value	**Sign**	**MSBP**	.....		**LSBP**
14	+	1	1	1	0
3	+	0	0	1	1
5	+	0	1	0	1
1	+	0	0	0	1
-2	-	0	0	1	0
7	+	0	1	1	1
-4	-	0	1	0	0
12	+	1	1	0	0
8	+	1	0	0	0

Moreover, let us assume that every bit-plane and the sign information is received with error after transmission over a noisy channel. In this case, the value of the received quantized coefficients is the following:

Table 4.3. Bit-plane representation of quantized coefficients after corruption.

Sign	MSBP	.....		LSBP	Coefficient value after errors
-	0	0	0	1	**-1**
-	1	1	0	0	**-12**
-	1	0	1	0	**-10**
-	1	1	1	0	**-14**
+	1	1	0	1	**13**
-	1	0	0	0	**-8**
+	1	0	1	1	**11**
-	0	0	1	1	**-3**
-	0	1	1	1	**-7**

The MSE of the code-block is then calculated as follows:

$$\frac{\left((14+1)^2+(3+12)^2+(5+10)^2+(1+14)^2+(-2-13)^2+\ldots+(8+7)^2\right)\cdot\Delta^2}{9}$$

$$=15^2\cdot\Delta^2 \tag{4.5}$$

which equals to:

$$(2^p-1)^2\cdot\Delta^2 \tag{4.6}$$

where p is the number of encoded bit-planes in the code-block (in this case, $p=4$). Hence, the term $M_{b,s}$ in Eq. (4.4) is equal to $(2^p-1)^2$ with p as the number of bit-

planes encoded in code-block (b, s). If lossy compression is used, some of the least significant coding passes are discarded and not included in the final bit-stream. In this case, $C_{n,b}$ is the nth quantized coefficient in code-block (b, s) represented by p completely encoded bit-planes, with d discarded least significant bit-planes. Taking into account the discarded bit-planes, d, Eq. (4.6) can be expressed as:

$$(2^{p} - 1 - \sum_{\tilde{p}=0}^{d-1} 2^{\tilde{p}})^2 \cdot \Delta^2 \tag{4.7}$$

where $\tilde{p}$ is the position of the discarded bit-plane with $\tilde{p} = 0$ for the least significant bit-plane and $\tilde{p} = p$ for the most significant bit-plane.

Considering the case of lossy compression, the $\overline{M}_{b,s}$ in a per-pixel basis can be written as:

$$\overline{M}_{b,s} = \frac{\left[(2^{-rb} \cdot \Delta) \cdot (2^{p} - 1 - \sum_{\tilde{p}=0}^{d-1} 2^{\tilde{p}}) \right]^2}{B_s} \tag{4.8}$$

Equation (4.8) provides a measure of the maximum effect of channel errors in a code-blocks. Note that Eq. (4.8) was derived under the assumption that no error resilient tools are employed in the source bit-stream. However, some JPEG2000 decoders [47] are designed to provide error concealment. These decoders make use of the error resilient tools to identify corrupted bit-planes in the code-blocks. When error concealment is used, after the occurrence of the first decoding error in a bit-plane of a code-block, the decoder assigns a value of zero to the subsequent bit-

planes, and as a result errors after the first error do not increase the MSE of the code-block any further. In this work, such a decoder is assumed. To illustrate this, let us assume the 3×3 code-block shown in Fig. 4.7 is received with error at the decoder side. If the first bit-error is found in the coding pass corresponding to the most significant bit-plane, the values of the reconstructed coefficients are as follows:

0	0	0
0	0	0
0	0	0

Figure 4.8. A 3×3-sample code-block after corruption in the most significant bit plane and error concealment.

In this case, the coefficients of the entire code-block are set to zero. Figure 4.9 shows the effect of error concealment in a corrupted image. Here, the gray-level Lena image was compressed as described in Fig. 4.6 with segmentation symbols, resetting and terminating the arithmetic coder. In this case, a single bit-error was simulated in the first byte of the initial code-block coded data of the bit-stream and error concealment was used at the decoder. Figure 4.9(a) shows the reconstructed image. Note the improvement in quality achieved by using error concealment. Fig. 4.9(b) shows the reconstructed image after multiple bit-errors with the first bit-error occurring in the first byte of the initial code-block coded data of the bit-stream. Notice that the quality of the reconstructed image is the same than the quality of the reconstructed image in Fig. 4.9(a) when error concealment is used.

(a) *PSNR=15.4338 dB* (b) *PSNR= 15.4338 dB*

Figure 4.9. Effect of single and multiple bit-errors in the initial code-block coded data when error concealment is used at the decoder. a) A single bit-error has been simulated in the first byte of the initial code-block coded data. b) Multiple bit-errors have been simulated in the initial code-block coded data, with the first bit-error occurring in the initial byte.

Results in Fig. 4.9 show that, when error concealment is used, the worst case is a bit-error occurring in the first bit-plane of a code-block. In this case, the coefficients of the entire code-block are set to zero. Hence, the MMSE for code-block (*b, s*), in a per pixel basis, is equal to the mean energy of the code-block, which can be calculated as follows:

$$\overline{\mathrm{M}}_{b,s} = \frac{2^{-2r_{b,s}}}{B_s} \cdot \frac{(\Delta_{b,s})^2}{N_{b,s}} \sum_{n=1}^{N_{b,s}} C_n^{\,2} \tag{4.9}$$

The $\overline{\mathrm{M}}_{b,s}$ in Eq. (4.9) provides a measure of the effect of channel errors in code-block (*b, s*) if error concealment is used. This measure will be used to assign channel

protection to the JPEG2000 bit-stream. Note that Eq. (4.9) is based on the assumption that orthogonal wavelets are used for decomposition. For non-orthogonal wavelets, Eq. (4.9) will provide an approximate result. In order to employ Eq. (4.9), the JPEG2000 bit-stream requires the addition of segmentation symbols and the reset and termination of the arithmetic coder, while the decoder must provide error concealment. The segmentation symbols are used to confirm the correct reception of the bit-planes and to determine the bit-planes that need to be set to zero.

4.3.2 Effect of Bit-errors in a Packet

When the coded data of a code-block is distributed across more than one layer, the packets that include the code-block for the first time are highly sensitive to errors. Bit-errors in these packets have a significant impact on the final MSE of the reconstructed image. The packets containing the rest of the coded data of a code-block have an important effect on the MSE of the reconstructed image only if the packet that contains the code-block for the first time is free of errors. Hence, in the proposed channel protection techniques a packet is protected according to the number of code-blocks that it includes for the first time. The protection assigned to a packet depends on the energy of the code-blocks as expressed in Eq. (4.9). Since each code-block is encoded independently from others, errors in one code-block will not propagate to other code-blocks. The MMSE in packet p with coded data from S sub-bands, each sub-band having B_s code-blocks, can be expressed as follows:

$$\overline{\mathrm{M}}_p \approx \sum_{s=1}^{S} \sum_{b=1}^{B_s} \left(\overline{\mathrm{M}}_{b,s} \cdot \psi(b) \right) \tag{4.10}$$

where $\psi(b)$ is 1 if the coded data for the *bth* code-block is included in packet p for the first time (otherwise it is zero), and $\overline{\mathrm{M}}_{b,s}$ is as given in Eq. (4.9). Note that the right side of Eq. (4.10) takes accounts of the MMSE of code-block (b, s) only if it is

included for the first time in packet p.

The effect of bit-errors in code-blocks and packets, as given in Eqs. (4.9) and (4.10) respectively, will be used to assign channel protection to the JPEG2000 bit-stream.

4.4 Equal Channel Protection

In this section, and equal channel protection technique for JPEG2000 images over Rayleigh fading channels is presented. The technique assigns the same protection to the all packets of the bit-stream according to the channel conditions. The channel protection is realized by means of a concatenation of a CRC outer coder and an inner RCPC coder.

4.4.1 Overall Distortion of the Reconstructed Image

The overall distortion of the image is typically defined as the MSE between the original and the received image and can be calculated as follows [48]:

$$D = E\{(I_o - I_q)^2\} + E\{(I_o - I_c)^2\} + E\{(I_o - I_q)(I_q - I_c)\} \qquad (4.11)$$

where I_o is the original image data, I_q is the noise-free reconstructed image, and I_c is the noisy reconstructed image. Under the assumption that the quantization noise is orthogonal to the channel noise, the third term in the right side of Eq. (4.11) becomes negligible. The overall distortion D can then be expressed as the summation of the individual distortions associated to each packet p multiplied by the probability of error, p_e, of the protected information. The probability of error p_e is the probability

of error of the protected information after channel protection for the current channel conditions. The channel protection is achieved by means of a concatenation of a CRC and RCPC code as described in section 3.2.5 and p_e is calculated for a Rayleigh fading channel as discussed in section 3.2.6.

For an image with K packets, the overall distortion can be computed as:

$$D = \sum_{p=1}^{K} \overline{\mathrm{M}}_p \cdot p_e \tag{4.12}$$

where $\overline{\mathrm{M}}_p$ is as given in Eq. (4.10). Note that if all the packets are protected and transmitted, $\sum_{p=1}^{K} \overline{\mathrm{M}}_p$ is the mean energy of the compressed image, denoted by $\bar{\varepsilon}_c$.

In order to accommodate the extra channel protection bits, some of the packets have to be discarded. Due to the embedded nature of the code stream, packets should be discarded in a sequential order starting with the last packet (if packet p is first discarded, the next packet to be discarded is always packet p-1). Each missing packet increases the overall distortion (see Eq. (4.12)) of the image. Taking into account the effect of discarded packets, Eq. (4.12) can be expressed as follows:

$$D = \left(\sum_{p=1}^{K} \overline{\mathrm{M}}_p \cdot \phi(p) \right) \cdot p_e + \sum_{p=1}^{K} m_p \cdot [1 - \phi(p)]$$

$$= \bar{\varepsilon}_c + \left(\sum_{p=1}^{K} \overline{\mathrm{M}}_p \, \phi(p) \right) \cdot p_e - \sum_{p=1}^{K} m_p \cdot \phi(p) \tag{4.13}$$

where m_p is the amount of MSE that will be added to the overall distortion if packet p is discarded, and $\phi(p)$ is 1 if the pth packet is included in the code-stream (otherwise it is zero). Note that $\sum_{p=1}^{K} m_p$ is, as well, the mean energy of the compressed image, $\bar{\varepsilon}_c$.

In the ECP technique, the distortion as given in Eq. (4.13) is minimized subject to the rate constraint, which is given below:

$$\frac{S_M}{R_M} + \frac{1}{R_p}\sum_{p=1}^{K} S_p \leq R_T \tag{4.14}$$

where R_M is the channel code rate for the main header, R_P is the channel code rate for the data packets, S_M is the number of bits in the main header, S_P is the number of source bits in each data packet. Note that the left side of Eq. (4.14) is the actual bit-rate whereas R_T is the specified overall bit-rate. Because of the importance of the main header, the ECP technique assigns the lowest channel-coding rate to the main header.

The minimum overall distortion in Eq. (4.13) is realized assuming a continuous code allocation for channel protection. However, the use of a family of RCPC codes results in discrete channel-coding rates. Furthermore, the amount of additional bandwidth obtained by discarding packets is also constrained by the discrete lengths of the JPEG2000 packets. These two constraints result in a sub-optimal solution. In this work, the optimization problem as represented by Eqs. (4.13) and (4.14) is solved by finding the minimum distortion for the specified overall bit-rate. The minimum distortion is found using the following algorithm:

1. Determine the overall distortion, D, by assigning the channel protection with the highest code rate to the current bit-stream and calculate the overall bit-rate R_T.
2. Check if R_T is approximately sufficiently close to the desired bit-rate; if so, store the overall distortion D in vector D_H and the channel code rate in vector H. Go to step 3. If the desired bit-rate has not been met, assign the next lower channel code rate to the bit-stream and repeat step 2. If the lowest channel code rate has already been assigned and the desired bit-rate has not been met, go to step 4.
3. Discard the last packet of the current bit-stream and repeat step 1. If there are no more packets to discard, go to step 4.
4. From vector H, select the channel code rate with the minimum associated distortion in D_H. This channel code rate is assigned to the current bit-stream.

4.5 Summary

Bit-errors can have a different impact on the JPEG2000 bit-stream depending on their position and occurrence. In this chapter, an analysis of the effect of bit-errors in code-blocks and packets was presented. First, a set of expressions in terms of the MSE was derived to describe the maximum impact of bit errors in the reconstructed image when no error resilience is employed. When error resilient tools are employed and the decoder provides error concealment, a set of expression was derived to describe the maximum MSE of code-blocks and packets. The latter set of expressions was used to develop an ECP technique. For a Rayleigh fading channel, the technique assigns the same channel protection to the entire JPEG2000 bit-stream according to the channel conditions. In addition, an algorithm was introduced to obtain the optimal channel protection.

Chapter 5

Unequal Channel Protection for JPEG2000 Images

The use of a single FEC code rate considers the compressed image data as equally important, resulting in ECP as discussed in Chapter 4. However, the hierarchical structure of the JPEG2000 bit-stream can be exploited to assign the channel protection in a more efficient way. Because the initial layers carry the most important bit-planes of the code-blocks used to compress the image, unequal channel protection can be applied to increase the robustness of the ECP for earlier portions of the bit-stream. In this way, the initial layers can receive more channel protection reducing or eliminating the channel protection assigned to the later layers. In this chapter, the use of unequal channel protection on JPEG2000 images is discussed. It is assumed a layered JPEG2000 bit-stream with one tile, one component, a PPM marker segment and a progression by resolution order. Two channel protection techniques are proposed. An unequal channel protection across the layers is first introduced, followed by an unequal channel protection across the packets. Both techniques are optimized for Rayleigh-fading channels and their performance is evaluated under different channel conditions. Similarly to the ECP, the proposed channel protection is achieved by means of a concatenation of a CRC and RCPC code.

5.1 Unequal Channel Protection across Layers

One first attempt to unequally protect the JPEG2000 bit-stream is to take advantage of the layered structure of the data and the error concealment tools of the decoder. When a multiple-quality-layer compression scheme is used (see section 4.2), the coded data of the code-blocks is distributed across a number of layers. Each layer successively and monotonically improves the image quality, and the decoder shall be able to decode the code-block contributions contained in each layer in sequence. The initial layers contain the most significant bit-planes of the code-blocks of all sub-bands at all resolution levels. As described in section 4.2, bit-errors in the initial layers result in a higher distortion of the reconstructed image.

Since the main idea behind unequal channel protection is to assign more redundancy to the layer containing the most important bit-planes of all code-blocks, we can assign all the channel protection to the initial layer, ignoring the subsequent layers. Let us remember that, if error concealment is used at the decoder, bit-errors after the occurrence of the initial bit-error do not degrade the quality of the reconstructed image any further. This UCP technique would work as long as all the code-blocks are included for the first time in the initial layer, however, this is not always true, since some of the most important bit-planes can be included in the later layers. To illustrate this, let us take the 512×512 gray-level Lena image and compress it using three levels of decomposition, a code-block size of 64×64, a precinct size of 512×512 and 10 quality layers. The resulting bit-stream has 40 packets, 4 for each layer, and 64 code-blocks (see Fig. 5.1).

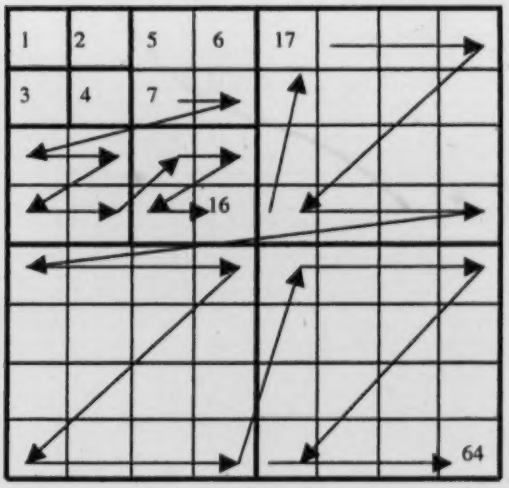


Figure 5.1. 64×64-sample code-blocks in the 512×512 gray-level Lena image with three levels of decomposition.

The first packet of each layer contains the coded data of code-block 1. The second packet of each layer contains the coded data of code-blocks 2,3,4. The third packet of each layer contains the coded data for code-blocks 5 to 16. The fourth packet of each layer contains the coded data for code-blocks 17 to 64.

The header of each packet contains information regarding the inclusion of a code-block for that packet. Table 5.1 summarizes the inclusion of code-blocks for every packet for the 10 different quality layers.

Table 5.1. Code-block contributions to data packets. Code-block numbers in bold indicate the packets and layers that include code-blocks for the first time. Packet 1 is the initial packet.

Layer	Packet 1	Packet 2	Packet 3	Packet 4
1	**1**	**2,4**	**5,7,8,9,12, 13**	**18,20,23,28-38,40,43-50,52,54-60,62,64**
2	1	2,**3**,4	5,**6**,7,8,9,**10**, 12,13	**17**,18,**19**,20,**21**,**22**,23,**24-27**,28-38,40,**41**,**42**,43-50,52,54-60,62,64
3	1	2,3,4	5,6,7,8,9,10, **11**,12,13,**14**	17-38,**39**,40-50,52,54-60,62,64
4	-	2,3,4	5,6,7,8,9,10, 11,12,13,14	17-50,**51**,54-64
5	-	3	5,6,7,8,9,10, 11,12,13,14,**15**	19-64
6	-	-	7,8,9,10,11, 12,13,14,15,**16**	28-57
7	-	-	13,14,15,16	28-50
8	-	-	11,12	31-50
9	-	-	-	31-46
10	-	-	-	43,44,45,**53**

The code-block numbers in bold in Table 5.1 show the packet and layer where a code-block is included for the first time. Note that some code-blocks may be included in the last layer.

Based on this observation, an unequal channel protection technique for JPEG2000

images is proposed. The UCP technique assigns channel protection to the layers according to their importance [33]. The channel protection is realized by means of a concatenation of a RCPC/CRC code.

The technique divides the JPEG2000 bit-stream into two parts, *M* and *L*. The main header, (including the tile-part header) comprises part *M*. The data packets contained in the layers comprise part *L*. The main header receives the lowest-rate channel protection while the layers receive channel protection according to the mean energy they contain.

The overall distortion *D* of the reconstructed image can then be expressed as the summation of the individual distortions associated to each layer multiplied by the probability of error, p_e^l, of the protected information in the layer. The distortion associated to a layer is computed by adding the distortions associated to each of the packets that comprise that layer. p_e^l is the probability of error of the protected information after applying the RCPC channel protection for the current channel conditions. Similarly to the ECP technique, p_e^l is calculated for a Rayleigh fading channel as described in section is 3.2.6.

For an image with *L* layers, the overall distortion can then be computed as:

$$D = \sum_{l=1}^{\widetilde{L}} \overline{\mathrm{M}}_l \cdot p_e^l \tag{5.1}$$

where $\overline{\mathrm{M}}_l$ is the distortion of layer *l*, and it is given as:

$$\overline{\mathrm{M}_l} = \sum_{p=1}^{P} \overline{\mathrm{M}}_p \tag{5.2}$$

with P as the total number of packets in layer l and $\overline{\mathrm{M}}_p$ the distortion associated to packet p as given in Eq. (4.10).

Similarly to the ECP technique, some of the packets of the final bit-stream have to be discarded in order to meet a target transmission rate R_T. Taking into account the distortion associated to the discarded packets, the overall distortion in Eq. (5.1), for an image with L layers with P packets in each layer, is given by:

$$D = \sum_{l=1}^{L}\left(\left(\sum_{p=1}^{P}\overline{\mathrm{M}}_{p,l}\cdot\phi(p)\right)\cdot p_e^l + \sum_{p=1}^{P} m_{p,l}\cdot\left[1-\phi(p)\right]\right) \tag{5.3}$$

where $m_{p,l}$ is the amount of MSE that will be added to the overall distortion if packet p in layer l is discarded, $\phi(p)$ is 1 if the *pth* packet is included in the code stream (otherwise it is zero) .

The distortion as given in Eq. (5.3) is minimized subject to the rate constraint:

$$\frac{S_M}{R_M} + \sum_{l=1}^{L}\frac{S_l}{R_l} \le R_T \tag{5.4}$$

where R_M is the channel code rate for the main header, R_l is the channel code rate for layer l, S_M is the number of bits in the main header and S_l is the number of source bits in layer l. In the same way than the ECP technique, the UCP technique protects the main header with the lowest channel-coding rate. Note that when packets are discarded, some layers are truncated and not fully transmitted. The channel protection assignment described by Eq. (5.3) corresponds to an unequal channel protection scheme across the layers, UCP-L hereafter.

5.2 Unequal Channel Protection across Packets

The UCP-L technique proposed in section 5.1 assigns the channel protection by exploiting the layered organization of the JPEG2000 bit-stream. Although the UCP-L technique protects the compressed image data in a more efficient way, it does not fully exploit the hierarchical structure of the bit-stream, specifically the importance of the position and inclusion of the most significant bit-planes in different packets of the final bit-stream. The information regarding the inclusion of a code-block can be used to assign the protection unequally within a layer. As described in section 2.7, within a layer, not all the packets contribute equally to the importance of the layer in terms of the overall distortion of the reconstructed image. The information in Table 5.1 shows that, within a layer, some packets may include more code-blocks for the first time than others. Based on this observation, a UCP protection across the packets is proposed (UCP-P hereafter) [34]. The UCP-P technique assigns the channel protection to the packets according to their importance. In this case, the technique divides the bit-stream into two parts, *M* and *P*. Part *M* contains the header and receives the lowest-rate channel protection. Part *P* is comprised by the data packets. Let us remember that bit-errors in the packets that include a code-block for the first time have a significant impact on the final MSE of the reconstructed image. Considering this, the overall distortion of the reconstructed image can be expressed as the summation of the individual distortions associated to each packet p multiplied by the probability of error P_e of every packet. P_e is calculated for a Rayleigh fading channel as described in section is 3.2.6.

For an image with K packets, this overall distortion is calculated as follows:

$$D = \sum_{p=1}^{K} \overline{M}_p \cdot P_e \tag{5.5}$$

with $\overline{\mathrm{M}}_p$ the distortion associated to packet p as given in Eq. (4.10).

As before, some packets need to be discarded in order to meet that target transmission bit-rate. Taking into account the discarded packets, the overall distortion of the reconstructed image for a bit-stream with K packets protected by means of the UCP-P technique is given by:

$$D = \sum_{p=1}^{K} \overline{\mathrm{M}}_p \cdot \mathrm{P}_e \cdot \phi(p) + \sum_{p=1}^{K} m_p \cdot [1 - \phi(p)]$$

$$= \sum_{p=1}^{K} \phi(p) \cdot \left[\overline{\mathrm{M}}_p \cdot \mathrm{P}_e - m_p \right] + \sum_{p=1}^{K} m_p$$

$$= \sum_{P=1}^{K} \phi(P) \cdot \left[\overline{\mathrm{M}}_P \cdot \mathrm{P}_e - m_P \right] + \bar{\varepsilon}_c \qquad (5.6)$$

where m_p is the amount of MSE that will be added to the overall distortion if packet p is discarded, $\phi(p)$ is 1 if the *pth* packet is included in the code stream (otherwise it is zero), and $\bar{\varepsilon}_c$ is the mean energy of the compressed image.

The optimal distortion is found minimizing Eq. (5.6) subject to the transmission rate constraint:

$$\frac{S_M}{R_M} + \sum_{p=1}^{K} \frac{S_p}{R_p} \leq R_T \qquad (5.7)$$

where R_M is the channel code rate for the main header, R_p is the channel code rate for packet p, S_M is the number of bits in the main header and S_p is the number of source bits in packet p. When a packet receives no channel protection, the channel code rate is equal to one. Equation (5.6) describes an unequal channel protection assignment across the packets.

5.3 Optimization of the Channel Protection

The optimization problem for the UCP-L and UCP-P techniques is solved by finding the points that lie on the lower convex hull of the rate-distortion plane corresponding to all possible channel-coding protection assignments. The points on the lower convex hull are found using an algorithm similar to the bit-allocation algorithm proposed in [49] by Westerink. Westerink's algorithm exploits the property of the convex hull to find all possible bit-allocations with the lowest distortion using a minimum number of calculations. Note that the UCP-P technique involves a larger number of calculations, since a JPEG2000 bit-stream has more packets than layers. This results in a greater complexity for the UCP-P technique, since more points on the lower convex hull have to be evaluated in order to obtain the optimal channel protection assignment. In this work, the minimum distortion is found using the following algorithm:

1. For a target transmission rate, determine the overall distortion D, of the current bit-stream using Westerink's algorithm employing only the packets or layers that include code-blocks for the first, and assuming an initial distortion and initial rate. The initial distortion is equal to the distortion incurred by the discarded packets. The initial rate is equal to the length of the protected main header plus the lengths of the data packets or layers that receive no channel protection (e.g. the data packets or layers that include no code-block code-

streams for the first time). Store the overall distortion D in vector D_H and the corresponding channel protection assignment in vector H.

2. Discard the last packet of the current bit-stream and repeat step 1. If there are no more packets to discard, go to step 3.
3. From vector H, select the channel protection assignment with minimum associated distortion in D_H.

This works assumes the use of no protocols for wireless transmission; therefore, the overhead associated with transmission protocols is ignored during the optimization process. Applying the previously described algorithm results in a set of optimal channel protection assignments that meet the target transmission rate R_T. For an image with K data packets after JPEG2000 compression, the set of optimal channel protection assignments can be expressed as:

$$H = \{h_n^K, h_n^{K-1}, h_n^{K-2}, ..., h_n^1, h_n^0\} \tag{5.8}$$

where h_n^t is a vector containing the channel protection assignment (in the form of channel-coding rates) with n protected packets or layers (according to the technique used, UCP-P or UCP-L respectively) and t transmitted packets for a target transmission rate R_T. Each of the channel protection assignments in H has an associated distortion, denoted as:

$$D_H = \{D_{h_n^K}, D_{h_n^{K-1}}, D_{h_n^{K-2}}, ..., D_{h_n^1}, D_{h_n^0}\} \tag{5.9}$$

The final optimal distortion, D_H^O, is then defined as the minimum distortion in D_H:

$$D_H^O = \min\{D_H\} \tag{5.10}$$

and the final optimal channel protection assignment, H^O, is the channel protection assignment in H with the final optimal distortion:

$$H^O = h_n^t \text{ if } D_{h_n^t} = D_O \tag{5.11}$$

where n denotes the number of protected packets or layers and t the number of transmitted packets. In practice, the set of channel protection assignments in H are sub-optimal due to the discrete channel-coding rates in a family of RCPC codes. Moreover, the amount of additional bandwidth obtained by discarding packets is also constrained by the discrete lengths of the JPEG2000 packets. Solutions close to the optimal case may still be achieved by making the JPEG2000 data packets as short as possible (e.g. using several layers) and by generating a wide range of RCPC channel-coding rates.

5.4 Simulation Results

In this section, the performance evaluation of the ECP, UCP-L and UCP-P techniques over a Rayleigh fading channel is presented. The 256 gray-level Lena image of size 512×512 pixels is used for simulations. This image is shown in Fig. 5.2.

Figure 5.2. Gray-level Lena image
used for transmission simulations.

Both lossless and lossy compressions (at three bit-rates: 0.5, 0.25, and 0.125 bpp) are considered. The overall transmission rate is set to the transmission rate of the image with no channel protection. The image is compressed using the Kakadu JPEG2000 codec software [47]. For lossless and lossy compression, the default reversible and irreversible wavelet transforms are used, respectively. The image is decomposed for five levels of decomposition, the size of the code block is set to 64×64, and the size of the precinct is set to 512×512. A code-block size of 64×64 samples is a good compromise between compression efficiency and resiliency to errors. With this configuration, the resulting bit-stream is comprised by six packets per layer. The arithmetic coder is terminated and reset after each coding pass, and segmentation symbols are added to the encoded bit-stream. At the packet level, the packet headers are moved to the main header in the PPM marker segment and SOP markers are added to the data packets. The received image information is decompressed using error concealment as discussed in section 4.3.1.

In order to obtain a solution close to the optimal case, the JPEG2000 packets are made as short as possible using several layers. For losseless compression, the coded data is divided into 20 quality layers. For lossy compression, 15 quality layers are used for a rate of 0.5 bpp, while 10 quality layers are used for a rate of 0.25 and 0.125 bpp.

Table 5.2 shows the code-block inclusion information for the JPEG2000 compressed gray-level image at the different bit-rates. Note that not all the packets include code-block code-stream for the first time.

Table 5.2. Code-block inclusion information for the JPEG2000 compressed gray-level Lena image. Packet 1 is the initial packet of the JPEG2000 bit-stream.

Bit-rate	Packets with code-block data for the first time
Lossless compression at 4.411 bpp (20 layers)	1,2,9,15,21,28,34,40,41,47,53,59, 60,66,71,72,73,78,84,90,96
Lossy compression at 0.5 bpp (15 layers)	1,2,8,14,15,21,27,34,40,46,47,53, 59,65,71,72,77,78,84,90
Lossy compression at 0.25 bpp (15 layers)	1,2,3,9,16,22,28,29,35,41,47,53, 54,59,60
Lossy compression at 0.125 bpp (10 layers)	1,2,8,15,21,28, 40,46,47,53, 59

In order to obtain different rates, the convolutional mother code of rate ¼ and generator matrix $g = [23\ 35\ 27\ 33]$, in octal notation, is punctured with a period of 8. The decoding process is performed using hard-decision decoding with the Viterbi algorithm. The puncturing matrices and RCPC codes obtained from the mother code are listed in Table 5.3.

Table 5.3. Puncturing matrices and RCPC codes.

Mother code [23 35 27 33]			
Rate r	Puncturing Matrix	Rate r	Puncturing Matrix
4/5	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	4/11	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&0&1&1&1&0\\0&0&0&0&0&0&0&0\end{bmatrix}$
2/3	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&0&1&0&1&0&1&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	1/3	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&0&0&0&0&0&0\end{bmatrix}$
4/7	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&0&1&1&1&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	4/13	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&0&0&0&1&0&0&0\end{bmatrix}$
1/2	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	2/7	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&0&1&0&1&0&1&0\end{bmatrix}$
4/9	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	4/15	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&0&1&1&1&0\end{bmatrix}$
4/10	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&0&0&1&1&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$	1/4	$\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1\end{bmatrix}$

An illustration of the behavior of the RCPC codes listed in Table 5.3 is presented in Fig. 5.3. These results show the bound on bit-error probability, P_b, as given in Eq. (3.42) for each channel-coding rate as a function of the $\overline{SNR}$ for a Rayleigh fading channel with hard decision decoding and ideal interleaving. Since a varying channel-coding rate is considered, the channel $\overline{SNR}$ is described by E_S / N_0 rather than $E_b / N_0 = E_S / rN_0$, with r as the channel-coding rate. The uncoded bit-error

probability is shown in Fig 5.3 as well. At low $\overline{SNR}$'s, when the bound on bit-error probability exceeds the uncoded bit-error probability, the uncoded bit-error probability is used instead of the bound.

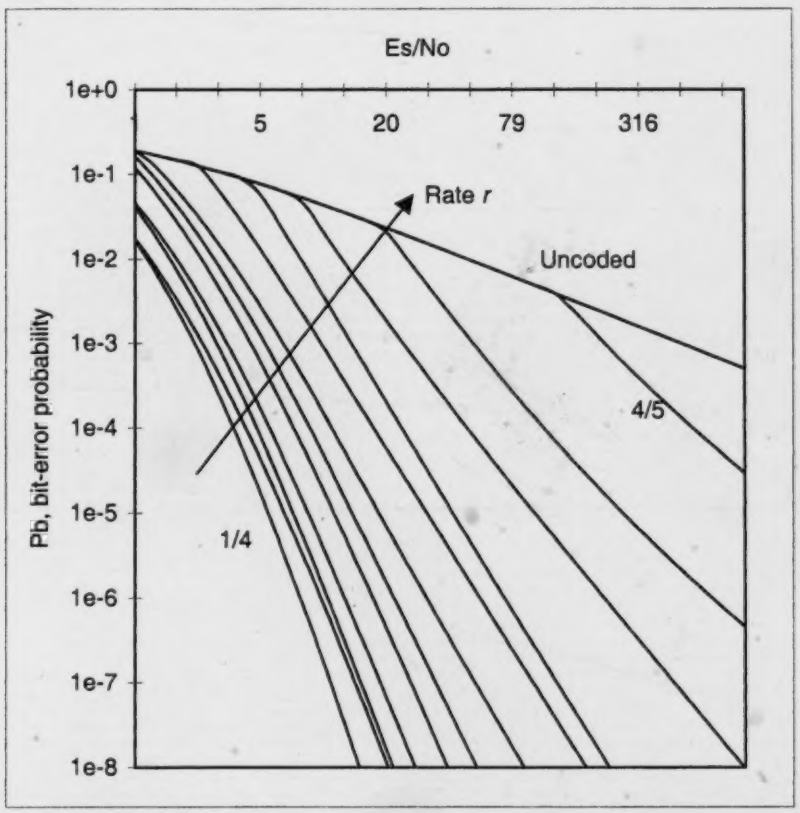


Figure 5.3. Probability of bit-error versus E_s / N_0 for a family of RCPC codes derived from the mother code of rate *R=1/4* and generator matrix *g=[23 35 27 33]*. The results are for a Rayleigh fading channel with hard decision decoding an ideal interleaving.

In all three techniques, the main header and the data packets to be protected are divided into blocks of *N* bits. Generally, the total number of source bits in the main header or a data packets cannot be evenly divided and the last block may have a size

less than *N*. In this work, *N* was set to 384 bits. Each block is first protected by an outer 16-bit CRC code defined by the polynomial 210421 (in octal notation), followed by an inner RCPC code. No puncturing matrix is used to protect the blocks in the main header. Before transmission, a convolutional interleaver of depth 60 interleaves the protected information. Figs. 5.4, 5.5 and 5.6 show a block diagram of the channel encoder for the ECP, UCP-L and UCP-P techniques, respectively.

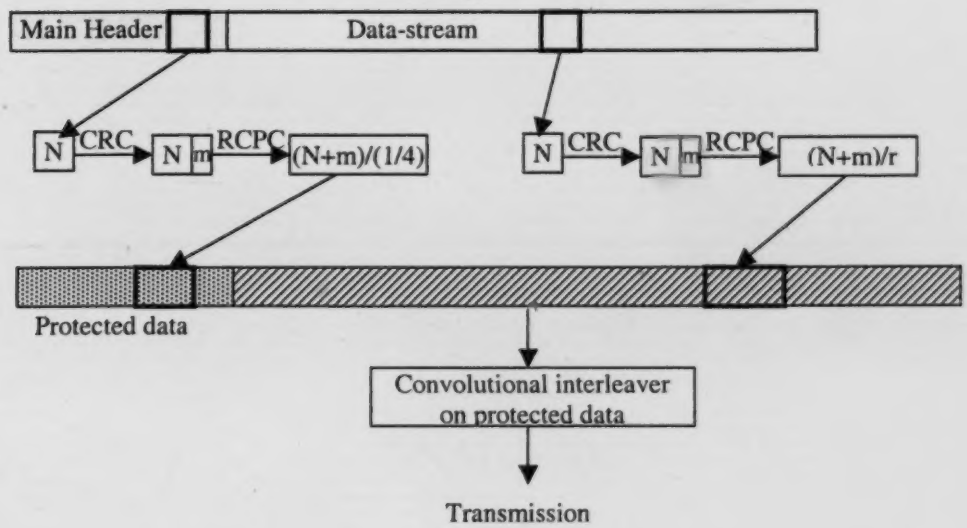


Figure 5.4. Block diagram of the ECP coder. N=384 bits, m=16 bits, r is the RCPC rate.

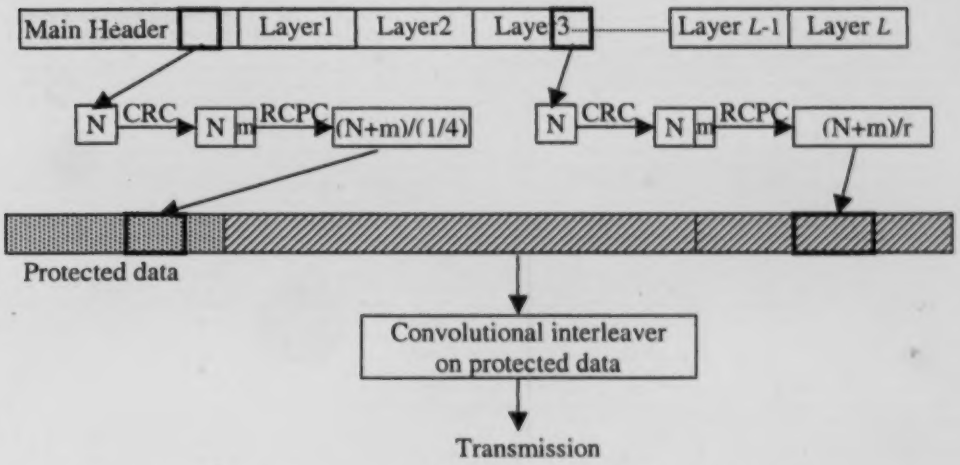


Figure 5.5. Block diagram of the UCP-L coder. N=384 bits, m=16 bits, r is the RCPC rate.

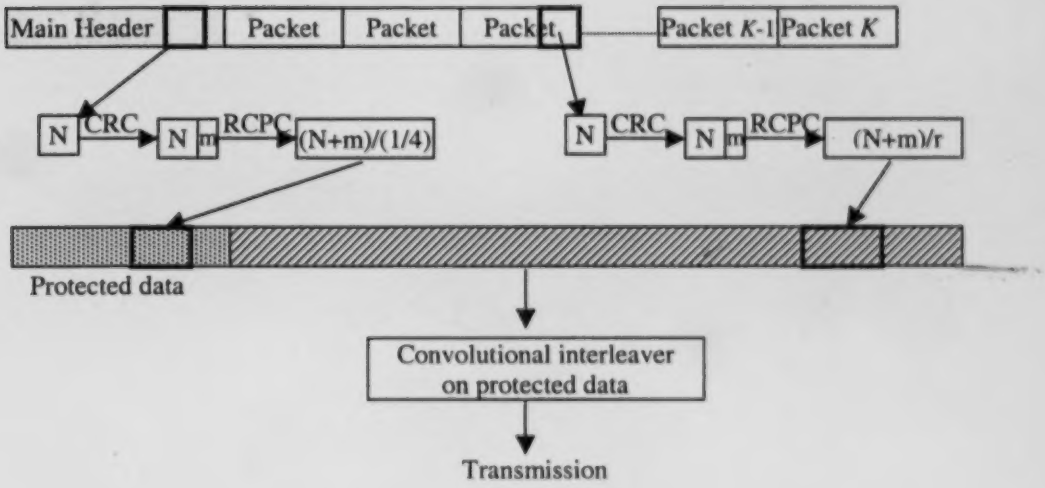


Figure 5.6. Block diagram of the UCP-P coder. N=384 bits, m=16 bits, r is the RCPC rate.

The information regarding the channel coding rate and number of protected packets is assumed to be common knowledge to both the encoder and decoder and no side information needs to be transmitted.

In order to know the theoretical performance of the channel protection techniques, the overall distortion of the reconstructed image has been plotted for different channel $\overline{SNR}$'s (Eqs. 4.13, 5.3 and 5.6). Figs. 5.7-5.9 show the results for the lossless case and lossy case at 0.5, 0.25, and 0.125 bpp, for the gray-level Lena image with a mobile speed of 3.6 Km/h. This speed corresponds to a person walking.

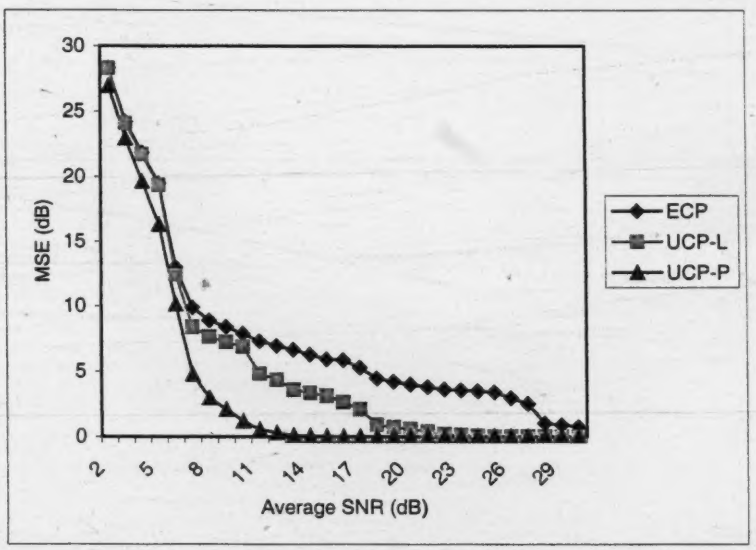
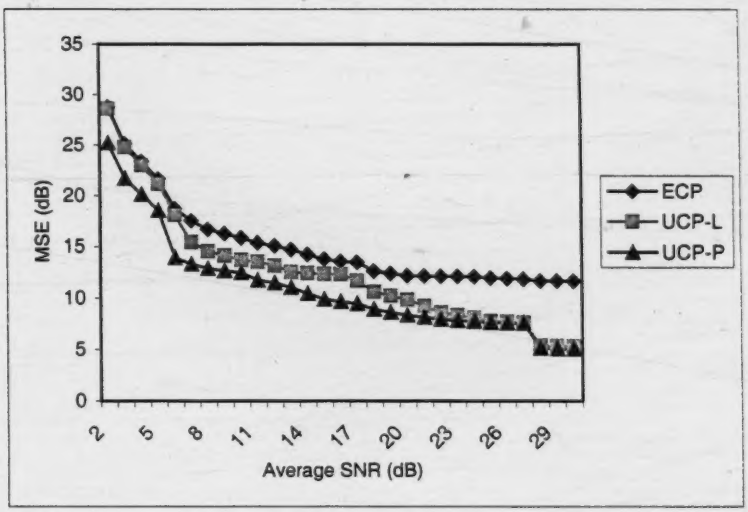
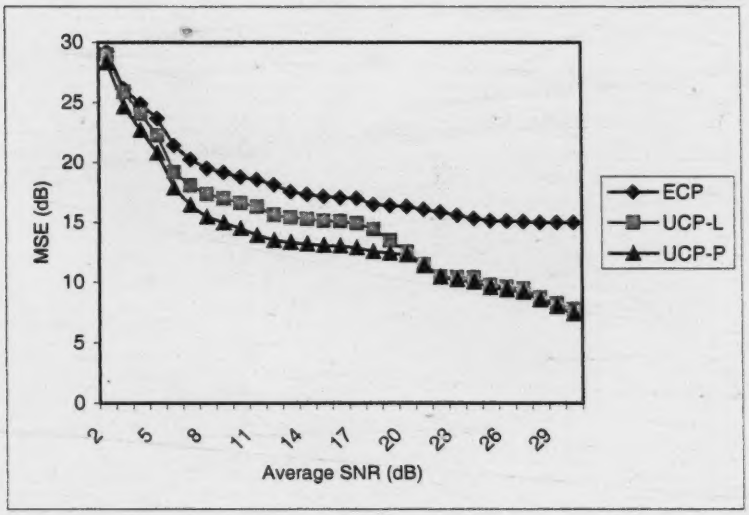


Figure 5.7. Theoretical distortion of the reconstructed gray-level Lena image at lossless compression for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 Km/h. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.



(a)



(b)

Figure 5.8. Theoretical distortion of the reconstructed gray-level Lena image at (a) 0.5 bpp and (b) 0.25bpp for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 Km/h. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

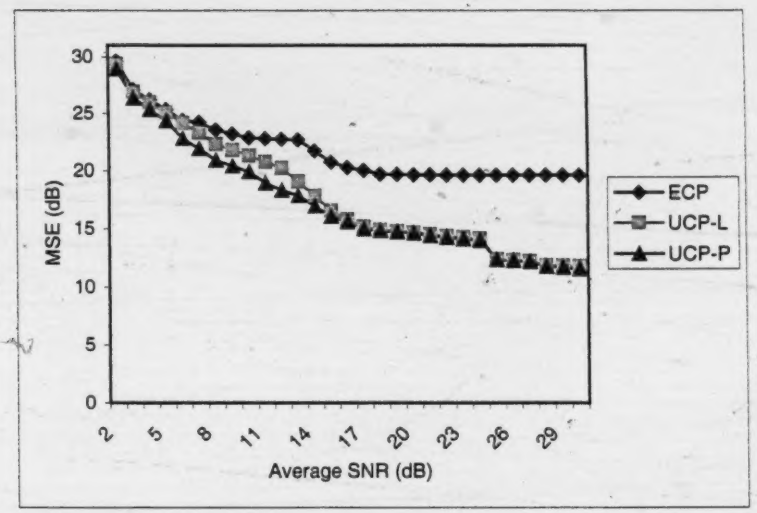


Figure 5.9. Theoretical distortion of the reconstructed gray-level Lena image at 0.125bpp for different $\overline{SNR}$'s (Rayleigh fading channel) with a mobile speed of 3.6 *Km/h*. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

It is apparent that the theoretical distortion of the reconstructed images is about 2dB MSE better for the UCP-P technique compared to the UCP-L technique and about 5 dB MSE better compared to the ECP technique over the range $10\text{dB} < \overline{SNR} < 22\text{dB}$. This proves the robustness of the UCP-P technique. At low $\overline{SNR}$ (<10 dB), a small improvement is seen mainly because RCPC codes are not capable to handle frequent long burst of errors. A better performance can be obtained by increasing the convolutional interleaver depth, but there is a penalty in the total number of bits transmitted as initial "zero" bits in the memory of the interleaver have to be flushed. However, at low $\overline{SNR}$ the UCP-P still provides a better performance.

Similarly, at large $\overline{SNR}$ (>22 dB), the improvement between the UCP-P and the UCP-L is small, mainly because the effect of bit-errors on the reconstructed image is

not as important as the effect of discarding packets. When the channel conditions improve, the UCP-P and the UCP-L techniques are able to reduce the channel coding rate and include more packets in the final bit-stream, reducing the overall distortion. On the other hand, the length constraint imposed by the protected information makes the ECP technique fail to achieve this, resulting in a higher distortion of the reconstructed image.

At low transmission rate (0.125 bpp), it is seen that the performance of the UCP-P technique is similar to that of the UCP-L at most $\overline{SNR}$'s. This suggests that the data packets at this transmission rate are short enough to efficiently discard packets and protect the information unequally across the layers and across the packets. In all cases, the minimum distortion is lower bounded by the distortion associated with discarding the necessary packets to accommodate the channel-protected main header, and upper bounded by the mean energy of the compressed image, $\overline{\varepsilon}_c$:

$$m_{p_n} \le D \le \overline{\varepsilon_c} \tag{5.12}$$

where D is the distortion of the reconstructed image and m_{p_n} is the distortion associated with discarding the n-last packets in order to accommodate the channel protection for the main header.

The robustness of the channel protection techniques with different mobile speeds is evaluated over four different channel conditions with a carrier frequency of 900 *MHz* and a data rate of 15 *kbits/sec*. The mobile speeds used are 3.6 *km/h*, 48.28 *km/h*, and 96.56 *km/h*. For comparison purpose, the image with no channel protection (NCP hereafter) and only the error resilient tools used for the channel protection techniques has been transmitted as well. Tables 5.4-5.12 show the channel coding schemes obtained by minimizing the overall distortion for the ECP, UCP-L and UCP-P

techniques for each channel condition and transmission rate.

Table 5.4. Channel coding rates for the Lena gray-level image with channel protection by means of the ECP technique. NTP is the number of transmitted packets. CR is the RCPC coding rate.

			Channel conditions Average SNR (dB) of the channel			
			10 dB	15 dB	20 dB	25 dB
Lossless 4.411 bpp 20 layers	*v*=3.6 *km/h*	NTP	108	107	113	113
		CR	4/9	1/2	4/5	4/5
	v=48.28 *km/h*	NTP	101	107	112	113
		CR	1/3	1/2	2/3	4/5
	v=96.56 *km/h*	NTP	101	107	112	113
		CR	1/3	1/2	2/3	4/5
Lossy 0.5 bpp 15 layers	*v*=3.6 *km/h*	NTP	73	77	81	81
		CR	1/2	2/3	4/5	4/5
	v=48.28 *km/h*	NTP	70	77	81	81
		CR	4/9	2/3	4/5	4/5
	v=96.56 *km/h*	NTP	69	76	81	81
		CR	4/10	4/7	4/5	4/5
Lossy 0.25 bpp 10 layers	*v*=3.6 *km/h*	NTP	40	45	46	46
		CR	1/2	2/3	4/5	4/5
	v=48.28 *km/h*	NTP	38	45	46	46
		CR	4/10	2/3	4/5	4/5
	v=96.56 *km/h*	NTP	38	44	46	46
		CR	4/10	4/7	4/5	4/5
Lossy 0.125 bpp 10 layers	*v*=3.6 *km/h*	NTP	43	43	43	43
		CR	4/5	4/5	4/5	4/5
	v=48.28 *km/h*	NTP	43	43	43	43
		CR	4/5	4/5	4/5	4/5
	v=96.56 *km/h*	NTP	43	43	43	43
		CR	4/5	4/5	4/5	4/5

Table 5.5. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at lossless compression. Twenty quality layers comprise the final bit-stream. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the layers.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	108	114	118	118
	CR	[1/3 1/3 1/3 1/3 1/3 1/3 1/3 1/3 4/10 4/11 4/10 4/11 4/10 4/10 4/9 4/10]	[4/9 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]	[1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]	[1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]
v=48.28 *km/h*	NTP	107	113	118	118
	CR	[2/7 2/7 4/13 4/13 4/13 1/3 1/3 1/3 1/3 1/3 1/3 1/3 1/3 1/3 4/10 4/11]	[1/3 4/11 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/10 4/10 4/10 4/9 4/9 4/9 4/9]	[4/9 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]	[1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]
v=96.56 *km/h*	NTP	107	113	118	118
	CR	[2/7 2/7 4/13 4/13 4/13 1/3 1/3 1/3 1/3 1/3 1/3 1/3 1/3 1/3 4/10 4/11]	[1/3 1/3 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/10 4/10 4/10 4/9 4/9 4/9 4/9]	[4/9 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]	[1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 4/7 1/2 4/7 1/2 4/7 4/7 4/7 4/7]

Table 5.6. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.5 bpp. Fifteen quality layers comprise the final bit-stream. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the layers.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	77	82	83	88
	CR	[4/13 4/10 4/10 4/9 4/9 4/10 4/9 4/9 1/2 4/7 4/7 1/2 4/7]	[4/9 4/7 1/2 4/7 4/7 4/7 4/7 4/7 4/7 2/3 2/3 2/3 2/3 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 4/5 4/5 4/5 4/5 4/5 4/5 4/5 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 2/3 2/3 4/5 8/8 8/8 8/8 8/8 8/8 4/5]
v=48.28 *km/h*	NTP	77	82	83	88
	CR	[4/13 4/10 4/10 4/9 4/9 4/10 4/9 4/9 1/2 4/7 4/7 1/2 4/7]	[4/9 4/7 1/2 4/7 4/7 4/7 4/7 4/7 4/7 2/3 2/3 2/3 2/3 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 4/5 4/5 4/5 4/5 4/5 4/5 4/5 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 2/3 2/3 4/5 8/8 8/8 8/8 8/8 8/8 4/5]
v=96.56 *km/h*	NTP	76	81	82	88
	CR	[2/7 4/10 1/3 4/10 4/10 4/11 4/10 4/10 4/9 4/9 1/2 4/9 4/5]	[4/10 1/2 4/9 1/2 1/2 1/2 1/2 4/7 4/7 4/7 2/3 4/7 2/3 4/5]	[1/2 4/7 1/2 4/7 4/7 4/7 4/7 4/7 4/7 2/3 2/3 2/3 2/3 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 2/3 2/3 4/5 8/8 8/8 8/8 8/8 8/8 4/5]

Table 5.7. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.25 bpp. Ten quality layers comprise the final bit-stream. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the layers.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	46	51	52	54
	CR	[4/11 4/9 4/9 1/2 1/2 4/7 4/7 4/5]	[1/2 2/3 2/3 2/3 2/3 2/3 2/3 2/3 4/5]	[4/7 2/3 2/3 2/3 2/3 4/5 4/5 4/5 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 8/8 8/8 8/8]
v=48.28 *km/h*	NTP	45	47	52	54
	CR	[4/13 4/10 4/10 4/9 4/9 1/2 4/7 4/5]	[4/9 4/7 4/7 4/7 4/7 4/7 2/3 2/3]	[4/7 2/3 2/3 2/3 2/3 4/5 4/5 4/5 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 8/8 8/8 8/8]
v=96.56 *km/h*	NTP	45	47	52	54
	CR	[4/13 4/10 4/10 4/9 4/9 1/2 4/7 4/5]	[4/9 4/7 4/7 4/7 4/7 4/7 2/3 2/3]	[4/7 2/3 2/3 2/3 2/3 4/5 4/5 4/5 4/5]	[4/7 2/3 2/3 2/3 2/3 2/3 8/8 8/8 8/8]

Table 5.8. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-L technique at 0.125 bpp. Ten quality layers comprise the final bit-stream. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the layers.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	51	51	51	51
	CR	[2/3 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]
v=48.28 *km/h*	NTP	51	51	51	51
	CR	[2/3 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]
v=96.56 *km/h*	NTP	51	51	51	51
	CR	[2/3 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]	[4/5 8/8 4/5 8/8 8/8 4/5 8/8 4/5 4/5]

Table 5.9. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at lossless compression. 20 quality layers with 120 packets comprise the final bit-stream. Packets No. 1,2,9,15,21,28,34,40,41, 47,53,59,60,66,71,72,73, 78,84,90,96 received channel protection. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the protected packets.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	118	119	119	119
	CR	[1/4 2/7 4/13 1/3 1/3 4/13 1/3 4/11 4/11 4/11 4/10 4/10 4/9 4/10 4/9 4/10 4/9 4/9 1/2 1/2]	[4/13 4/13 1/3 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 1/2 1/2 1/2 1/2]	[1/3 1/3 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]	[1/3 4/11 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]
v=48.28 *km/h*	NTP	117	119	119	119
	CR	[1/4 1/4 2/7 4/13 4/13 4/13 1/3 1/3 1/3 1/3 4/10 4/11 4/9 4/10 4/9 4/10 4/9 4/9 1/2 1/2]	[4/13 4/13 1/3 4/11 4/11 4/11 4/11 4/11 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]	[4/13 1/3 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]	[1/3 4/11 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]
v=96.56 *km/h*	NTP	117	119	119	119
	CR	[1/4 1/4 2/7 4/13 4/13 4/13 1/3 1/3 1/3 1/3 4/10 4/11 4/9 4/10 4/9 4/10 4/9 4/9 1/2 1/2]	[2/7 4/13 1/3 4/11 4/11 1/3 4/11 4/11 4/11 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]	[4/13 1/3 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]	[1/3 4/11 4/11 4/11 4/11 4/11 4/11 4/10 4/10 4/11 4/10 4/10 4/9 4/10 1/2 4/10 4/9 1/2 1/2 1/2]

Table 5.10. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.5bpp. 15quality layers with 90 packets comprise the final bit-stream. Packets No. 1,2,8,14,15,21,27,34, 40,46,47,53, 59,65,71,72, 77,78,84,90 received channel protection. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the protected packets.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	87	87	88	88
	CR	[2/3 4/5 4/5 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 4/5 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]
v=48.28 *km/h*	NTP	80	87	88	88
	CR	[2/3 2/3 2/3 2/3 2/3 4/5 4/5 2/3 4/5 4/5 4/5 4/5 4/5 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 4/5 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]
v=96.56 *km/h*	NTP	80	86	88	88
	CR	[2/3 2/3 2/3 2/3 2/3 4/5 4/5 2/3 4/5 4/5 4/5 4/5 4/5 8/8 8/8 8/8 8/8 8/8]	[2/3 4/5 4/5 4/5 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 4/5 8/8 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]

Table 5.11. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.25 bpp. 10 quality layers with 60 packets comprise the final bit-stream. Packets No. 1,2,3,9,16,22,28,29,35,41,47,53, 54,59,60 received channel protection. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the protected packets.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	56	57	57	58
	CR	[2/3 4/5 4/5 4/5 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]
v=48.28 *km/h*	NTP	53	56	57	58
	CR	[2/3 2/3 2/3 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 4/5 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]
v=96.56 *km/h*	NTP	53	56	57	58
	CR	[2/3 2/3 2/3 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[2/3 4/5 4/5 4/5 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 4/5 8/8 4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]	[4/5 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8 8/8]

Table 5.12. Protection schemes for the gray-level Lena image with channel protection by means of the UCP-P technique at 0.125 bpp. 10 quality layers with 60 packets comprise the final bit-stream. Packets No. 1,2,8,15,21,28, 40,46,47,53, 59 received channel protection. NTP is the number of transmitted packets. CR is the RCPC coding rate scheme for the protected packets.

		Channel conditions Average SNR (dB) of the channel			
		10 dB	15 dB	20 dB	25 dB
v=3.6 *km/h*	NTP	50	51	51	51
	CR	[2/3 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 8/8 8/8 8/8]	[4/5 4/5 8/8 8/8 8/8 8/8 8/8]
v=48.28 *km/h*	NTP	50	50	50	50
	CR	[2/3 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]
v=96.56 *km/h*	NTP	50	50	50	50
	CR	[2/3 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]	[4/5 4/5 8/8 4/5 8/8 8/8 8/8]

Some important observations can be made from the results in Tables 5.4-5.12.

i) Depending on the channel conditions and the energy distribution of the image across the layers and packets, it is sometimes preferable to assign less protection to the protected layers or packets and transmit more packets rather than increase the protection and discard packets. Note that, in some cases, some layers or packets receive no channel protection. This suggests

that the amount of energy contained in those layers or packets is not important. Moreover, the bound on bit-error probability exceeds the uncoded bit-error probability, as shown in Fig. 5.3, and no channel protection is chosen over the a RCPC code.

ii) The UCP-L and UCP-P techniques easily adapt to the channel conditions. Notice how the channel protection decreases as the channel conditions improve, while the number of transmitted packets increases. Moreover, because of the short length of the data packets, the UCP-P technique assigns the channel protection efficiently when the channel conditions change.

iii) As expected from the evaluation of the theoretical performance, the ECP technique fails to adapt when the channel conditions change mainly due to the length constraint imposed by the size of the data to be protected. In some cases, this technique assigns the same level of protection to the bit-stream for different channel conditions.

Each channel condition has been tested with 500 independent trials. Tables 5.13-5.15 show the average PSNR of the received images after FSK transmission over a Rayleigh fading channel with NCP, ECP, UCP-L and UCP-P. When the header contains too many bit-errors, the JPEG2000 decoder may not be able to decode the received bit-stream. The decoding probability in Tables 5.13-5.15 provide the probability of successful decoding the bit-stream.

Table 5.13. Average PSNR (in dB) and decoding probability (Dec. Prob.) for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh fading channel with a mobile speed of 3.6 *km/h*. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

Transmission Method		Average SNR (dB) of the channel							
		10 dB		15 dB		20 dB		25 dB	
		PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.
Lossless 20 layers	NCP	19.061	0.630	22.755	0.800	27.193	0.840	28.885	0.890
	ECP	24.463	0.808	27.787	0.950	31.980	0.998	33.298	1.00
	UCP-L	25.775	0.858	29.532	0.988	34.093	0.996	35.050	1.00
	UCP-P	26.881	0.836	31.593	0.990	36.274	0.998	37.011	1.00
Lossy 0.5 bpp 15 layers	NCP	19.509	0.750	20.553	0.810	27.017	0.880	29.680	0.905
	ECP	24.996	0.880	27.465	0.990	32.983	0.990	33.487	0.998
	UCP-L	26.622	0.850	31.616	0.970	34.696	0.990	37.776	1.00
	UCP-P	29.460	0.890	34.098	0.970	37.004	0.990	38.055	0.998
Lossy 0.25 bpp 10 layers	NCP	18.885	0.686	21.877	0.770	23.797	0.830	24.707	0.904
	ECP	23.598	0.870	25.845	0.990	27.946	0.990	28.103	1.00
	UCP-L	25.234	0.830	29.486	0.998	33.992	0.990	35.093	0.998
	UCP-P	27.261	0.840	32.440	0.980	35.016	0.970	36.566	0.998
Lossy 0.125 bpp 10 layers	NCP	19.846	0.730	20.589	0.804	22.033	0.814	22.276	0.920
	ECP	23.343	0.840	24.459	0.970	25.032	0.990	25.980	1.00
	UCP-L	26.654	0.860	29.046	0.970	30.454	0.998	30.865	1.00
	UCP-P	27.543	0.890	30.540	0.980	30.904	0.990	31.154	1.00

Table 5.14. Average PSNR (in dB) and decoding probability (Dec. Prob.) for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh fading channel with a mobile speed of 48.28 *km/h*. NCP is no channel protection. ECP is the equal channel protection technique UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

Transmission Method		Average SNR (dB) of the channel							
		10 dB		15 dB		20 dB		25 dB	
		PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.
Lossless 20 layers	NCP	18.705	0.590	18.887	0.740	25.180	0.790	26.192	0.810
	ECP	20.021	0.735	21.626	0.915	28.621	0.982	28.906	1.00
	UCP-L	22.799	0.750	26.721	0.910	32.430	0.986	35.937	0.998
	UCP-P	24.920	0.730	29.900	0.930	33.555	0.980	36.629	0.996
Lossy 0.5 bpp 15 layers	NCP	15.332	0.600	19.795	0.720	23.849	0.846	25.176	0.900
	ECP	20.331	0.725	25.304	0.940	28.087	0.980	32.270	0.998
	UCP-L	24.874	0.710	29.801	0.965	31.261	0.978	33.110	0.998
	UCP-P	26.992	0.690	32.397	0.950	34.905	0.988	36.719	0.996
Lossy 0.25 bpp 10 layers	NCP	15.960	0.580	18.663	0.724	21.898	0.844	23.797	0.894
	ECP	20.276	0.710	23.960	0.957	26.342	0.978	27.220	0.998
	UCP-L	21.847	0.762	26.551	0.955	30.985	0.965	34.581	0.998
	UCP-P	24.332	0.700	30.919	0.960	34.109	0.970	35.935	0.998
Lossy 0.125 bpp 10 layers	NCP	17.308	0.610	19.706	0.760	21.973	0.880	22.063	0.914
	ECP	21.239	0.704	22.844	0.918	24.075	0.966	24.158	0.998
	UCP-L	23.794	0.718	28.181	0.924	28.466	0.968	29.125	0.994
	UCP-P	24.575	0.710	29.163	0.920	29.924	0.990	30.361	0.998

Table 5.15. Average PSNR (in dB) and decoding probability (Dec. Prob.) for the JPEG2000 512 × 512 gray-level Lena image after transmission over a Rayleigh-fading channel with a mobile speed of 96.56 *km/h*. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

Transmission Method		Average SNR (dB) of the channel							
		10 dB		15 dB		20 dB		25 dB	
		PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.	PSNR	Dec. Prob.
Lossless 20 layers	NCP	16.322	0.626	16.758	0.704	20.383	0.790	23.293	0.810
	ECP	20.019	0.730	21.219	0.890	26.765	0.980	27.211	0.996
	UCP-L	21.402	0.720	26.721	0.880	31.678	0.982	33.486	0.998
	UCP-P	23.292	0.740	28.581	0.900	32.664	0.970	34.514	0.994
Lossy 0.5 bpp 15 layers	NCP	15.085	0.520	18.978	0.710	21.192	0.830	25.070	0.870
	ECP	19.931	0.690	23.816	0.940	26.738	0.974	31.747	0.990
	UCP-L	23.264	0.660	27.986	0.960	30.969	0.974	32.854	0.988
	UCP-P	26.113	0.700	31.384	0.950	33.837	0.980	35.228	0.988
Lossy 0.25 bpp 10 layers	NCP	15.547	0.534	16.518	0.680	18.885	0.790	21.002	0.814
	ECP	19.686	0.730	22.139	0.945	25.023	0.960	26.735	0.990
	UCP-L	21.626	0.712	25.751	0.930	29.673	0.955	31.185	0.980
	UCP-P	24.277	0.720	29.001	0.930	33.753	0.949	34.479	0.980
Lossy 0.125 bpp 10 layers	NCP	15.532	0.590	16.308	0.710	19.068	0.824	20.504	0.920
	ECP	18.431	0.712	21.351	0.910	23.278	0.960	23.922	0.984
	UCP-L	22.939	0.710	27.278	0.912	27.434	0.966	28.181	0.976
	UCP-P	23.010	0.708	28.076	0.916	28.582	0.964	29.766	0.982

Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 10 dB, 15 dB, 20dB and 25 dB with a mobile speed of 3.6 *km/h* are shown in Figs. 5.10-5.13. The images show the average quality for the three channel protection techniques and the case of no channel protection.

Figure 5.10. Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 10 dB for the gray-level Lena image. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

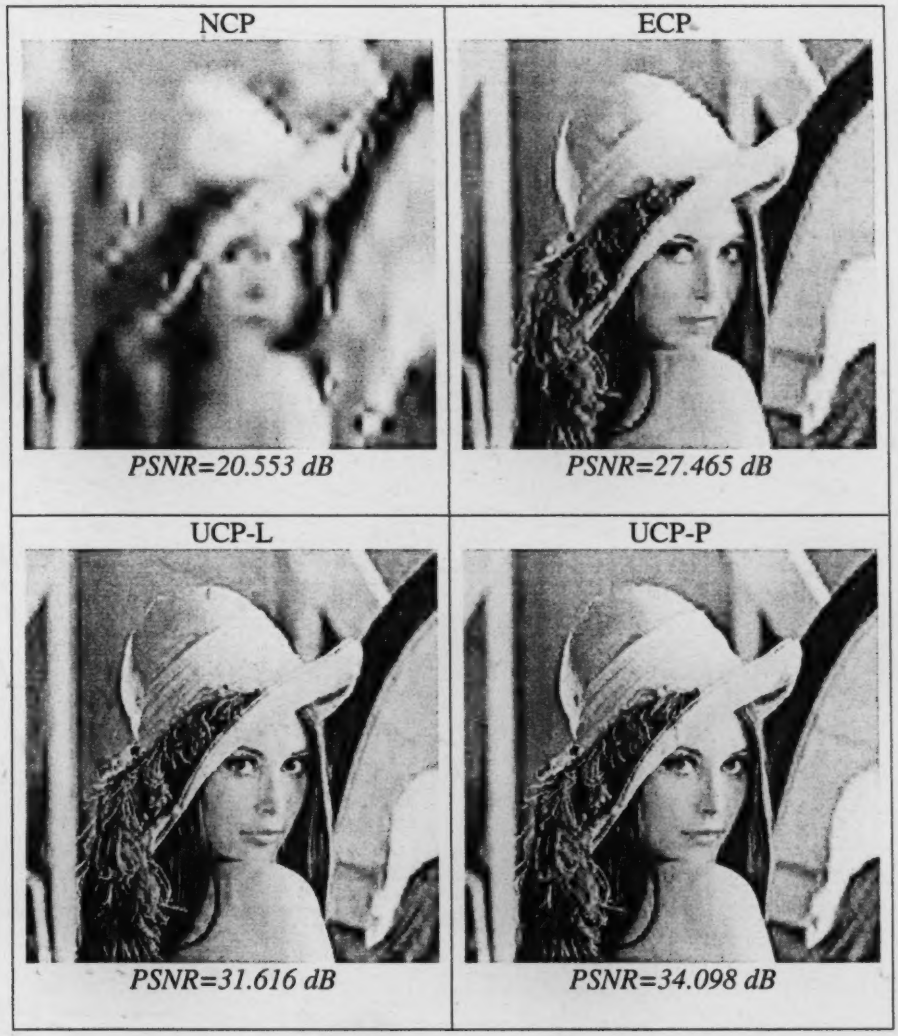


Figure 5.11. Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 15 dB for the gray-level Lena image. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

Figure 5.12. Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 20 dB for the gray-level Lena image. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

Figure 5.13. Visual results at 0.5bpp over a channel with $\overline{SNR}$ = 25 dB for the gray-level Lena image. NCP is no channel protection. ECP is the equal channel protection technique. UCP-L is the unequal channel protection across the layers technique. UCP-P is the unequal channel protection across the packets technique.

According to results in Figs. 5.10-5.13, the UCP techniques provide the best visual results. Note that the images protected by means of the ECP technique tend to be blurred due to the discarded packets. The effect of discarding packets is even evident under good channel conditions. The effect of error concealment can be appreciated in all received images, as the visual quality of the reconstructed images gracefully degrades as the conditions worsen. Under good channel conditions, the performance of the UCP channel protection techniques is similar; however, an improvement on the visual quality for the UCP-P techniques can still be appreciated. Notice the quality of the circled areas in Fig. 5.13.

Regarding the average quality of the reconstructed images, it is observed in Tables 5.13-5-15 that the ECP technique provides a considerable improvement on the probability of decoding the received information and the quality of the reconstructed images, over the NCP technique at all channel conditions. This shows that the error resilient tools are not always capable to deal with bit-errors in the code-stream. By assigning channel protection to the bit-stream, the error resiliency is improved even when packets are discarded. It is important to notice that when no channel protection is applied to the bit-stream, errors in the main header may result in a lower quality of the reconstructed image even if the data packet are received correctly. Let us remember that the main header contains the information used to decode the data packets. This information, which is included in the PPM marker segment, is highly sensitive to transmission errors.

The UCP-L improves even further the error resiliency of the bit-stream by unequally protecting the layers. Simulation results show an improvement of around 3 dB PSNR compared to the ECP technique. The improvement is constant across the four different channel conditions and transmission rates, and it is even greater when the channel conditions improve. This demonstrates the capability of the UCP-L technique to adapt to good channel conditions, where it is sometimes preferable to

reduce the channel protection and transmit more packets.

As expected from the evaluation of the theoretical performance, the UCP-P technique provides the best performance across all channel conditions. Results show an improvement of about 1.5 dB PSNR over the UCP-L technique and about 4 dB PSNR over the ECP technique. The UCP-P technique not only takes into account the effect of discarding packets and the effect of channel errors on the packets, but it also considers the inclusion of the code-blocks in the final bit-stream. This results in a better protection scheme according to the channel conditions. Note that under good channel conditions, the performance of the UCP-P technique tends to be similar to that of the UCP-L technique for the same number of transmitted packets. These results agree with the curves obtained from the evaluation of the theoretical performances.

5.5 Summary

Two UCP techniques were proposed in this chapter. The UCP-L and UCP-P techniques exploit the hierarchical structure of the JPEG2000 bit-stream and unequally protect the compressed data across the layers and across the packets, respectively. In both UCP techniques, the channel protection is assigned optimally according to the channel conditions and the mean energy contained in the different layers and packets. A performance evaluation of the techniques was presented for a Rayleigh fading channel and comparisons were made with the case of no channel protection and equal channel protection. Simulation results showed an improvement on the quality of the reconstructed images when the UCP techniques were employed. The UCP-P provides the best performance across different transmission rates and channel conditions.

Chapter 6

Conclusions and Future Work

When dealing with wireless channels, the bandwidth constraints imposed by channel and the noise introduced to the compressed data make the problem of image communications difficult. In order to improve the quality of the reconstructed images, error resilient tools can be applied to the compressed bit-stream. If the error resilient tools are not powerful enough to meet a desired target quality due to several bit-errors in the compressed data, channel protection may still be used. Typical channel protection techniques add the same amount of redundancy to the entire bit-stream. Although, ECP techniques can improve the resilience to errors of the compressed data, they are not suitable for the JPEG2000 bit-stream. This thesis investigated the problem of channel protection for JPEG2000 images over Rayleigh fading channels. Two different channel protection techniques were proposed, UCP-L and UCP-P. These unequal channel protection techniques add redundancy to the bit-stream unequally across the layers and packets, respectively. The techniques take advantage of the error resilient tools proposed in the JPEG2000 standard and assumed the use of error concealment methods at the decoder side. In all the techniques, the channel protection was assigned optimally based on the mean energy contained in the data packets of the JPEG2000 bit-stream.

The UCP-L and UCP-P technique proved to provide the better performance compared to an equal channel protection scheme. Among the three channel protection techniques (ECP, UCP-L and UCP-P), the UCP-P technique demonstrated to perform better under different channel conditions.

Although the proposed techniques provided a superior performance, there are still several issues related to the robust transmission of JPEG2000 images over noisy channels. Some potential future work in this area is:

i) Investigating the case of unequal channel protection within a data packet. Let us remember that each packet contains a number of individual code-block bit-streams. Unequal channel protection can be applied within a packet adding more redundancy to the code-blocks with the most energy.

ii) Investigating the case of unequal channel protection within the main header. The different marker segments in the main header can be channel protected unequally according to the importance of the information they contain.

iii) Investigating the use of diversity techniques when the channel conditions are considerably poor. The use of diversity techniques provides the advantage of reducing the amount of channel protection since several copies of the transmitted compressed data are available at the decoder side.

iv) Determining the performance of the proposed channel protection techniques with more powerful FEC codes, such as turbo-codes.

v) Determining the number of decomposition levels and data packet size to obtain the best-reconstructed image quality using the channel protection techniques proposed in this thesis. Similarly, determining the optimal number of layers in order to achieve the best performance.

vi) Testing the proposed channel protection techniques with more complex channel models.

vii) Extending the channel protection techniques to the case of transmission of color images.

viii) Similarly, extending the channel protection techniques to other multimedia coding algorithms where the compressed bit-stream has a hierarchical structure. An example of such coding algorithms is the MPEG-4 standard for audio-visual information.

References

[1] A. Perkis, "On the importance of error resilience in visual communications over noisy channels," *Circuits Systems Signal Processing,* vol. 20, no. 3, pp. 415-446, 2001.

[2] S. S. Hemamai, "Robust image communication over wireless channels," *IEEE Communications Magazine,* pp. 120-124, November 2001.

[3] J. Liang and R. Talluri, "Tools for robust image and video coding in JPEG2000 and MPEG4 standards," *Proc. IS&T-SPIE Visual Commun. Image Processing 1999,* vol. 3654, pp. 40-51.

[4] B. D. Pettijohn, M. W. Hoffman and K. Sayood, "Joint source/channel coding using arithmetic codes," *IEEE Transactions on Communications,* vol.49, no. 5, pp. 826-836, May 2001.

[5] H. Li and C. W. Chen, "Robust image transmission with bidirectional synchronization and hierarchical error correction," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 11, no. 11, pp. 1183-1187, November 2001.

[6] C. W. Chen and Z. Sun, "Image transmission over noisy channels with variable-coefficient fixed-length coding scheme," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9 no. 5, pp. 680-682, August 1999.

[7] A. Abrardo, M. Barni and A. Garzelli, "Improved group-of-block synchronization for robust transmission of H.263 video over slow-fading channels", *Journal of Electronic Imaging*, vol. 10, pp. 1016-1024, October 2001.

[8] J. Wen and J. Villasenor, "Reversible variable length codes for efficient and robust image and video coding," *IEEE Proceedings of the Data Compression Conference*, Snowbird, Utah, pp. 471-480, April, 1998.

[9] P. C. Cosman, J. K. Rogers, P. G. Sherwood and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp.982-993, June 2000.

[10] S. Lin, D. Costello and M. J. Miller, "Automatic repeat request error control scheme," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5-16, December 1984.

[11] V. Weerackody and W. Zeng, "ARQ schemes with switched antenna diversity and their applications in JPEG image transmission," *IEEE Globecom*, vol. 3, pp. 1915-1919, 1995.

[12] K. R. Narayanana and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Communications Letters*, vol. 1, pp. 49-51, March 1997.

[13] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Transactions on Communications*, vol 48, no. 6 pp. 948-958, June 2000.

[14] R. C. Luo and M. G. Kay, "A tutorial on multisensor integration and fusion," *16th Annual Conference of IEEE Industrial Electronics Society, IECON '90*, vol.1 pp. 707-722, 1990.

[15] L. C. Ramac and P. K. Varshney, "A wavelet domain diversity method for transmission of images over wireless channels," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 891-898, June 2000.

[16] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on partitioning in hierarchical trees," *IEEE Transactions on Circuits Syst. Video Technol.,* vol. 6, pp. 243-250, June 1996.

[17] A. Guyader, E. Fabre, C. Guillemot and M. Roberts, "Joint source-channel turbo decoding of entropy-coded sources," *IEEE Journal on Selected Areas in Communications,* vol. 19, no. 9, pp. 1680-1696, September 2001.

[18] K. Sayood, H. H. Out and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Transactions on Communications,* vol. 48, no. 5, pp. 787-794, May 2000.

[19] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE Journal on Selected Areas in Communications,* vol. 10, no. 5, pp. 926-943, June 1992.

[20] T. Fingscheidt, T. Hindelang, R. V. Cox and N. Seshadri, "Joint source-channel (de-)coding for mobile communications," *IEEE Transactions on Communications,* vol. 50, no. 2, pp. 200-212, February 2002.

[21] J. Cai and C. W. Chen, "Robust joint source-channel coding for image transmission over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 10, no. 6, pp. 962-966, September 2000.

[22] K. P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable-length encoded sources: the BSC case," *IEEE Transactions on Communications,* vol. 49, no. 12, pp. 2052-2055, December 2001.

[23] M. Bystrom and J. W. Modestino, "Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel," *IEEE Journal on Selected Areas in Communications,* vol. 19, no. 6 pp. 880-890, June 2000.

[24] A. Alavi, R. Link, and S. Kallel, "Adaptive unequal error protection for subband image coding," *IEEE Transactions on Broadcasting*, vol. 46, no. 3, pp. 197-205, September 2000

[25] M. G. Strintzis, "Optimal Pyramid and subband decomposition for hierarchical coding of noisy and quantized images," *IEEE Transactions on Image Processing*, vol. 7, no.2 pp. 155-166, February 1998.

[26] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447-457, 1998.

[27] M. J. Ruf and J. W. Modestino, "Operational rate-distortion performance for joint source and channel coding for images," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 305-320, March 1999.

[28] H. Man, F. Kossentini and M. J. T. Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 95-108, February 1999.

[29] P. G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Transactions on Communications*, vol. 46, no. 12, pp. 1555-1559, December 1998.

[30] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, December 1993.

[31] I. H. Witten, A. Moffat, T. C. Bell, *Managing Gigabytes. Compressing and Indexing Documents and Images.* Morgan Kaufmann Publishers, 1999.

[32] JPEG 2000 Part I: Final Draft International Standard (ISO/IEC FDIS15444-1), ISO/IECJTC1 /SC29/WG1 N1855, Aug. 2000.

[33] V. Sanchez and M. K. Mandal, "Robust transmission of JPEG2000 images over noisy channels," *Digest of Technical Papers ICCE 2002*, pp. 80-81, June 2002.

[34] V. Sanchez and M. K. Mandal, "Adaptive unequal channel protection for JPEG2000 images," submitted to *The III Indian Conference on Computer Vision, Graphics and Image Processing*, December 2002.

[35] A. Skodras, C. Christopoulos and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Processing Magazine,* pp. 36-58, September 2001.

[36] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing,* vol. 9, pp. 1158-1170, July 2000.

[37] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG2000," *IEEE Signal Processing Magazine,* pp. 22-35, September 2001.

[38] A. M. Michelson and A. L. Levesque, *Error-control Techniques for Digital Communications.* John Wiley & Sons, 1985.

[39] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. Journal*, p. 1253, September 1960.

[40] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, pp. 2316-2323, Aug. 1995.

[41] John G. Proakis, *Digital Communications*. McGraw-Hill, 1983.

[42] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[43] A. J. Viterbi, "Convolutional codes and their performance in communications system," *IEEE Transactions on Communications Technology,* vol. COM-19, No. 5, pp. 751-772, Oct 1971.

[44] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun*, vol. 36, pp. 389-400, April 1988.

[45] I. Moccagatta, S. Soudagar, J. Liang and H. Cheng, "Error-Resilient coding in JPEG2000 and MPEG4," *IEEE Journal on Selected Areas of Communication,* vol. 18, no. 6, pp. 899-914, June 2000.

[46] R. C. Gonzalez, P. Wintz, *Digital Image Processing.* Second Ed., Reading, Ma. Addison-Wesley, 1987.

[47] Kakadu software implementation of the JPEG 2000 standard. http://www.kakadusoftware.com

[48] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless noisy channels," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 850-860, June 2000.

[49] P. H. Westerink, J. Biebomond, and D. E. Boekee, "An optimal bit allocation algorithm for sub-band coding," *Proc. IEEE ICASSP,* pp. 757-760, 1998.